Intrinsic Dimension Estimation: Relevant techniques and a Benchmark Framework

P. Campadelli E. Casiraghi C. Ceruti A. Rozza

July 19, 2017

Abstract

When dealing with datasets comprising high-dimensional points, it is usually advantageous to discover some data structure. A fundamental information needed to this aim is the minimum number of parameters required to describe the data while minimizing the information loss. This number, usually called intrinsic dimension, can be interpreted as the dimension of the manifold from which the input data are supposed to be drawn.

Due to its usefulness in many theoretical and practical problems, in the last decades the concept of intrinsic dimension has gained considerable attention in the scientific community, motivating the large number of intrinsic dimensionality estimators proposed in literature. However, the problem is still open since most techniques cannot efficiently deal with datasets drawn from manifolds of high intrinsic dimension and non-linearly embedded in higher dimensional spaces.

This paper surveys some of the most interesting, widespread used, and advanced state-of-the-art methodologies. Unfortunately, since no benchmark database exists in this research field, an objective comparison among different techniques is not possible. Consequently, we suggest a benchmark framework and apply it to comparatively evaluate relevant state-of-the-art estimators.

keywords: Survey, Intrinsic dimension estimation, Manifold learning, Dimensionality reduction, Benchmark framework.

1 Introduction

Since the 1950s, the rapid pace of technological advances allows to measure and record increasing amounts of data, motivating the urgent need to develop dimensionality reduction systems to be applied on real datasets comprising high-dimensional points.

To this aim, a fundamental information is provided by the **intrinsic dimension** (`id`) defined by Bennet [3] as the minimum number of parameters needed to generate a data description by maintaining the "intrinsic" structure characterizing the dataset, so that the information loss is minimized.

More recently, a quite intuitive definition employed by several authors in the past has been reported by Bishop in [6], p. 314, where the author writes that “a set in D dimensions is said to have an `id` equal to d if the data lies entirely within a d-dimensional subspace of $\Re^D$”.

Though more specific and different `id` definitions have been proposed in different research fields [21, 90, 91], throughout the pattern recognition literature the presently prevailing `id` definition views a point set as a sample set uniformly drawn from an unknown smooth (or locally smooth) manifold structure, eventually embedded in an higher dimensional space through a non-linear smooth mapping; in this case, the `id` to be estimated is the manifold’s **topological dimension**.

Due to the importance of `id` in several theoretical and practical application fields, in the last two decades a great deal of research effort has been devoted to the development of effective `id` estimators. Though several techniques have been proposed in literature, the problem is still open for the following main reasons.

At first, it must be highlighted that though Lebesgue’s definition of topological dimension [61] (see Section 3.2) is quite clear, in practice its estimation is difficult if only a finite set of points is available. Therefore, `id` estimation techniques proposed in literature are either founded on different notions of dimension (e.g. fractal dimensions Section 3.2.1) approximating the topological one, or on various techniques aimed at preserving the characteristics of data-neighborhood distributions, which reflect the topology of the underlying manifold. Besides, the estimated `id` value markedly changes as the scale used to analyze the input dataset changes [62] (see an example in Figure 1), and being the number of available points practically limited, several methods underestimate `id` when its value is sufficiently high (namely $\mathtt{id} \geqslant 10$). Other serious problems arise when the dataset is embedded in higher dimensional spaces through a non-linear map. Finally, the too high computational complexity of most estimators makes them unpractical when the need is to process datasets comprising huge amounts of high-dimensional data.

In this work, after recalling the application domains of interest, we survey some of the most interesting, widespread used, and advanced `id` estimators. Unfortunately, since each method has been evaluated on different datasets, it is difficult to compare them by solely analyzing the results reported by the authors. This highlights the need of a benchmark framework, such as the one proposed in this work, to objectively assess and compare different techniques in terms of robustness w.r.t. parameter settings, high dimensional datasets, datasets being characterized by an high `id`, and noisy datasets.

The paper is organized as follows: in Section 2 the usefulness of the `id` knowledge is motivated and interesting `id` application domains profitably exploiting it are recalled; in Section 3 we survey notable state-of-the-art `id` estimators, grouping them according to the employed methods; in Section 4 we summarize mostly used experimental settings, we propose a benchmark framework, and we employ it to objectively assess and compare relevant `id` estimators; in Section 5 conclusions and open research problems are reported.

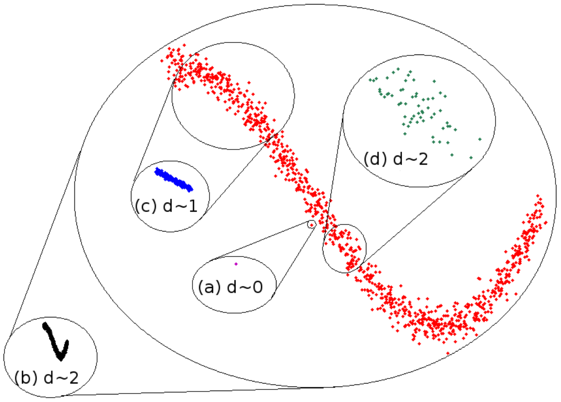


Figure 1: At very small scales (a) the dataset seems zero-dimensional; in this example, when the resolution is increased until including all the dataset (b) the id looks larger and seems to equal the embedding space dimension; the same effect happens when it is estimated at noise level (d); the correct id estimate is obtained at an intermediate resolution

2 Application Domains

In this section we motivate the increasing research interest aimed at the development of automatic id estimators, and we recall different application contexts where the knowledge of the id of the available input datasets is a profitable information.

In the field of pattern recognition, the id is one of the first and fundamental informations required by several dimensionality reduction techniques [128, 43, 42, 127, 121], which try to represent the data in a more compact, but still informative, way to reduce the "curse of dimensionality" effects [2]. Furthermore, when using an auto-associative neural network to perform a nonlinear feature extraction, the id value d can suggest a reasonable value for the number of hidden neurons [63]. Indeed, a network with a single hidden layer of neurons with linear activation functions has an error function with a unique global minimum and, at this minimum, the network performs a projection on the subspace spanned by the first d principal components [56] estimated on the dataset (see 8.6.2 of [6]), being d the number of hidden neurons. Besides, according to statistical learning theory [120], the capacity and generalization capability of a given classifier 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 $d+1$ (see [120], pp. 156-158). Since the generalization error depends on the VC dimension, it follows that the generalization capability may depend on the id value d. Moreover, in [40] the authors mark that, in order to balance a classifier generalization ability and its empirical error, the complexity of the classification model should also be related to the id of the available dataset. Furthermore, since complex objects can be considered as structures composed by multiple manifolds that must be clustered to be processed separately, the knowledge of the local ids characterizing the considered object is fundamental to obtain a proper clustering [17].

These observations motivate applications employing global or local id estimates to discover some structure within the data. In the following we summarize or simply recall some interesting examples [45, 68, 15, 119, 18, 69, 51].

In [45] the authors introduce a fractal dimension estimator, called Correlation Dimension estimator (CD, see Section 3.2.1), and show that the id estimate it computes is a reliable approximation of the strange attractor dimension in chaotic dynamical systems.

In the field of gene expression analysis, the work proposed in [68] shows that the id estimate computed by the nearest neighbor estimator (described in [92] and Section 3.2.2) is a lower bound for the number of genes to be used in supervised and unsupervised class separation of cancer and other diseases. This information is important since generally used datasets contain large number of genes and the classification results strongly depend on the number of genes employed to learn the separation criteria.

In [15], the authors show that id estimation methods being derived from the basis theory of fractal dimensions ([45, 62, 73], see Section 3.2.1), can be successfully used to evaluate the model order in signals and time series, which is the number of past samples required to model the time series adequately and is crucial to make reliable predictions. This comparative work employs fractal dimension estimators, since the domain of attraction of nonlinear dynamic systems has a very complex geometric structure, which could be captured by closely related studies on fractal geometry and fractal dimensions.

A noteworthy research work in the field of crystallography [119] employs the fractal CD estimator [45] followed by a correction method [16] that, according to the authors, “is needed because the CD estimator, to give correct estimations of the id, requires an unrealistically large number of points”. Anyway, the experimental results show that id is a useful information to be exploited when analyzing crystal structures. This study not only proves that id estimates are especially useful when dealing with practical tasks concerning real data, but also underlines the need to compute reliable estimates on datasets drawn from manifolds characterized by high id and embedded in spaces of much greater dimensionality.

The work of Carter [18] is very interesting and notable because it is one of the first considering that the input data might be drawn from a multi-manifold structure, where each sub-manifold has a (possibly) different id. To separate the manifolds, the authors compute local id estimates, by applying both a fractal dimension estimator (namely MLE [73], see Section 3.2.1) and a nearest neighbor-based estimator (described in [27, 28], see Section 3.2.2) on properly defined data neighborhoods. The authors then show that the computed local ids might be helpful for the following interesting applications: (1) “Debiasing global id estimates": the negative bias caused both by the limited number of available sample points and by the *curse of dimensionality* is reduced by computing global id estimates through a weighted average of the local ones, which assign greater importance to the points away from the boundaries. However the authors themselves note that this method is only applicable for data with a relatively low id, since in high dimensions the points lye nearby the boundaries [4]. (2) "Statistical Manifold Learning": the local id estimates are used to reduce the dimension of statistical manifolds [19], that is manifolds whose points represent a pdf. When this step is applied as the first step of document classification applications, and analysis of patients' samples acquired in the field of flow cytometry, it allows to obtain lower dimensional points showing a good class separation. (3) "Network Anomaly Detection": considering that the overall complexity of a router network is decreased when few sources account for a disproportionate amount of traffic, a decrease in the id of the entire network is searched for. (4) "Clustering": problems of data clustering and image segmentation are dealt with by assuming that different clusters and image patches belong to manifold structures characterized by different complexity (and ids).

In [69], to the aim of analyzing gene expression time series, the authors compute id estimates by comparing the fractal CD estimator and the Nearest Neighbor estimator (NN [92]). The results on both simulated and real data show that NN seems to be more robust than CD w.r.t. non-linear embeddings and the underlying time-series model.

In the field of Geophysical signal processing, hyperspectral images, whose pixels represent spectra generated by the combination of an unknown set of independent contributions, called endmembers, often require to estimate the number of endmembers. To this aim, the proposal in [51] is to substitute state of the art algorithms specifically designed to solve this task, with id estimators. After motivating the idea by describing the relation between the id of a dataset and the number of endmembers, the authors choose to experiment two fractal id estimators [45, 62] and a nearest neighbor-based one [37]. They obtain the most reliable results with the latest one after opportunely tuning the number of nearest neighbors to be considered.

Finally, other noteworthy examples of research works that profitably exploit id, and estimate it by usually applying fractal dimension estimators, concern financial time series prediction [104], biomedical signal analysis [24, 84, 32], analysis of ecological time series [53], radar clutter identification [48], speech analysis [110], data mining and low dimensional representation of (biomedical) time series [52], plant traits representation [70].

3 Intrinsic Dimension Estimators

In this section we survey some of the most notable, recent, and effective state-of-the-art id estimators, grouping them according to the main ideas they are based on.

Specifically, in Section 3.1 we describe **projective id estimators**, which basically process a dataset $\boldsymbol{P}_N \equiv \{\boldsymbol{p}_i\}_{i=1}^N \subseteq \Re^D$ to identify a somehow appealing lower dimensional subspace where to project the data and whose vector space dimension is viewed as the `id` estimate.

More recent projective `id` estimators exploit the assumption of datasets $\boldsymbol{P}_N \equiv \{\boldsymbol{p}_i\}_{i=1}^N \subseteq \Re^D$ being uniformly drawn from a smooth (or locally smooth) manifold $\boldsymbol{\mathcal{M}} \subseteq \Re^d$, embedded into a higher D-dimensional space through a non-linear map; this is also the basic assumption of all the other groups of methods, that will be referred to as **topological `id` estimators** (see Section 3.2) and **graph-based `id` estimators** (see Section 3.3).

We note that the taxonomy we are using to group the reviewed methods is different from the one, commonly used by several authors in the past (as an example, see [14]), that viewed methods as **global**, when `id` estimation is performed by considering a dataset as a whole, or **local**, when all the data neighborhoods are analyzed separately and an estimate is computed by combining all the local results. All the recent methods have abandoned the global approach since it is now clear that analyzing a dataset at its biggest scale cannot produce reliable results. They thus estimate the global `id` by somehow combining local `id`s. This way of proceeding comes from the assumption that the underlying manifold is locally smooth.

3.1 Projective `id` Estimators

The first projective `id` estimators introduced in literature explicitly compute the mapping that projects input points $\boldsymbol{P}_N \in \Re^D$ to the subspace $\boldsymbol{\mathcal{M}} \subseteq \Re^d$ minimizing the information loss [14, 73], and therefore view the `id` as the minimal number of vectors linearly spanning the subspace $\boldsymbol{\mathcal{M}}$. It must be noted that, since these methods were originally designed for exploratory data analysis and dimensionality reduction, they often require the dimensionality of $\boldsymbol{\mathcal{M}}$ (the `id` to be estimated) to be provided as input parameter. However, their extensions and variants include methodologies to automatically estimate `id`.

Most of the projective `id` estimators can be grouped into two main categories: projection techniques based on Multidimensional Scaling (`MDS`, [99, 100]) or its variants, which tend to preserve as much as possible pairwise distances among the data; projection techniques based on Principal Component Analysis (`PCA`, [56, 76]) and its variants, that search for the best projection subspace $\boldsymbol{\mathcal{M}}$ minimizing the projection error.

Some of the best known examples of `MDS` algorithms are `MDSCAL` [106, 107, 65, 67, 108, 66], Bennett's algorithm [3, 22], Sammon's mapping [103], Curvilinear Component Analysis (`CCA`) [31], `ISOMAP` [115] and Local Linear Embedding (`LLE` [101]). As shown by experiments reported in [115, 101] `ISOMAP` and variants of `LLE` compute the most reliable `id` estimates. We believe that their better performance is due to the fact that both `ISOMAP` and `LLE` have been the first projective methods based on the assumption that the input points are drawn from an underlying manifold, whose curvature might affect the precision of data neighborhoods computed by employing the Euclidean distance. However, as noted in [72, 59], these algorithms have shown to suffer of all the major drawbacks affecting MDS-based algorithms, which are too much tied by the preservation of the pairwise distance values. Besides, as highlighted in [28], ISOMAP as an id estimator, as well as other spectral based methods like PCA, relies on a specific estimated eigenstructure that may not exist in real data. Regarding LLE, it either requires the id value to be known in advance, or it may automatically estimate it by analyzing the eigenvalues of the data neighborhoods [94]; however, as outlined in [62, 73], id estimates computed by means of eigenvalue analysis are as unreliable as those computed by most PCA-based approaches. Moreover, in [76] it is noted that methods such as LLE, are based on the solution of a sparse eigenvalue problem under the unit covariance constraint; however, due to this imposed constraint, the global shape of the embedded data can not reflect the underlying manifold.

PCA [56, 76] is one of the most cited and well known projective id estimator, often used as the first step of several pattern recognition problems, to compute low dimensional representations of the available datasets. When PCA is used for id estimation, the estimate is the number of "most relevant" eigenvectors of the sample covariance matrix, also called principal components (PCs). Due to the promising dimensionality-reduction results, several PCA-based approaches, both deterministic and probabilistic, have been published. Among deterministic approaches, we recall the Kernel PCA (KPCA [105]), the local PCA (LPCA [41]) and its extensions to automatically select the number of PCs [122, 13]. We observe that the work presented in [13] is one of the first that estimates id by considering an underlying topological structure, and therefore applies LPCA on data neighborhoods represented by an Optimally Topology Preserving Map[1] (OTPM) built on clustered data. The authors of this method state that their approach is more efficient and less sensitive to noise w.r.t. the PCA-based approaches. However they do not show any experimental comparison and, besides, their algorithm employs critical thresholds and a data clustering technique whose result heavily influences the precision of the computed estimate [73].

The usage of a probabilistic approach has been firstly introduced by Tipping and Bishop in [116]. Considering that deterministic methodologies lack an associated probabilistic model for the observed data, their Probabilistic PCA (PPCA) reformulates PCA as the maximum likelihood solution of a specific latent variable model. PPCA and its extensions to both mixture and hierarchical mixture models have been successfully applied to several real problems; but they still provide an id-estimation mechanism depending on critical thresholds. This motivates its subsequent variants [34] and developments, whose examples are

[1] Given an input dataset $\boldsymbol{P}_N$, its OTPMs is usually computed through Topology Representing Networks (TRNs); these are unsupervised neural networks [82] developed to map $\boldsymbol{P}_N$ to a set of neurons whose learnt connections define proximities in $\boldsymbol{P}_N$. These proximities correspond to the optimal topology preserving Voronoi tessellation, and the corresponding Delaunay triangulation. In other words, TRNs compute connectivity structures that define and perfectly preserve the topology originally present in the data, forming a discrete path-preserving representation of the inner (topological) structure of the topological manifold underlying the dataset $\boldsymbol{P}_N$.

Bayesian PCA (BPCA [7]), and two Bayesian model order selection methods introduced in [97, 85]. In [9] the asymptotic consistency of id estimation by a (constrained) isotropic version of PPCA is shown with numerical experiments on simulated and real datasets.

While the aforementioned methods have been simply recalled since their id estimation results have shown to be unreliable [62, 73], in the following recent and promising proposals are described with more details.

The Simple Exponential Family PCA (SePCA [75]) has been developed to overcome the assumption of Gaussian-distributed data that makes it difficult to handle all types of practical observations, e.g. integers and binary values. SePCA achieves promising results by using exponential family distributions; however, it is highly influenced by critical parameter settings and it is successful only if the data distribution is known, which is often not the case, specially when highly non-linear manifold structures must be treated.

In [46] the authors propose the Sparse Probability PCA (SPPCA) as a probabilistic version of the Sparse PCA (SPCA [129]). Precisely, SPCA selects id by forcing the sparsity of the projection matrix, that is the matrix containing the PCs. However, based on the consideration that the level of sparsity is not automatically determined by SPCA, SPPCA employs a Bayesian formulation of SPCA, achieving sparsity by employing a different prior and automatically learning the hyper-parameter related to the constraint weight through Evidence Approximation ([8]-Section 3.5). The authors' results and also the results of the comparative evaluation proposed in [20] show that this method seems to be less affected by the problems of the aforementioned projective schemes.

An alternative method (MLSVD, [77]) applies Singular Value Decomposition (SVD), basically a variant of PCA, locally and in a multi-scale fashion to estimate the id characterizing D-dimensional datasets drawn from non-linearly embedded d-dimensional manifolds $\boldsymbol{\mathcal{M}}$ corrupted by Gaussian noise. Precisely, exploiting the same ideas of the theoretical PCA-based id estimator presented in [60], the authors note that the best way to avoid the effects of the curvature (induced by the non-linearity of the embedding) is to apply SVD locally, that is in hyperpheres $\boldsymbol{\mathcal{B}}(\boldsymbol{p}, r)$ centered on the data points $\boldsymbol{p}$ and having radius r. However, the choice of r is constrained by the following considerations: (1) r must be big enough to have at least $k \geq d$ neighbors, (2) r must be small enough to ensure that $\boldsymbol{\mathcal{M}} \cap \boldsymbol{\mathcal{B}}$ is linear (or at least smooth) (3) r must be big enough to ensure that the effect of noise are negligible. When these three constraints are met, the tangent space $T^d_{\boldsymbol{\mathcal{M}}}(\boldsymbol{p}, r)$, computed by applying SVD on the k neighbors, is a good approximation of the tangent space of $\boldsymbol{\mathcal{M}} \cap \boldsymbol{\mathcal{B}}$ and the number of its relevant eigenvalues correspond to the (local) id of $\boldsymbol{\mathcal{M}}$. To find a proper value for r, the authors propose a multi-scale approach that applies SVD on neighborhoods $\boldsymbol{\mathcal{B}}(\boldsymbol{p}, r_s)$ whose radius varies in a range $r_s \in \{r_L..r_H\}$. This allows to compute D scale-dependent, local singular values $\lambda_1(\boldsymbol{p}, r_s) \geq \ldots \geq \lambda_D(\boldsymbol{p}, r_s)$; using a least squares fitting procedure the SVs can be expressed as functions of r whose analysis allows to identify the range of scales $[r_{min}, ..., r_{max}]$ not influenced by either noise or curvature. Finally, in the range $r_s = [r_{min}, ..., r_{max}]$

the squared SVs are analyzed to get the id estimate $\hat{d}$ that maximizes the gap $\Delta(j) = \lambda_j(\boldsymbol{p}, r_s) - \lambda_{j+1}(\boldsymbol{p}, r_s)$ for the largest range of r_s. The proposed algorithm has been evaluated on unit d-dimensional hyperpheres and cubes embedded in $\Re^{100}$ and affected by Gaussian noise. The reported results are very good, while other ten well known methods [45, 73, 28, 49, 47, 23, 18] show that the ids estimated on the same datasets are unreliable also in the absence of noise.

3.2 Topological Approaches

Topological approaches for id estimation consider a manifold $\boldsymbol{\mathcal{M}} \subseteq \Re^d$ embedded in a higher dimensional space $\Re^D$ through a proper (locally) smooth map $\phi : \boldsymbol{\mathcal{M}} \to \Re^D$, and assume that the given dataset is $\boldsymbol{P}_N = \{\boldsymbol{p}_i\}_{i=1}^N = \{\phi(\boldsymbol{x}_i)\}_{i=1}^N \subset \Re^D$, where $\boldsymbol{x}_i$ are independent identically distributed (i.i.d.) points drawn from $\boldsymbol{\mathcal{M}}$ through a smooth probability density function (pdf) $f : \boldsymbol{\mathcal{M}} \to \Re^+$.

Under this assumption the id to be estimated is the manifold's topological dimension, defined either through the firstly proposed Brouwer's Large Inductive Dimension [12] or the equivalent Lebesgue's Covering Dimension [55]. Since Brouwer's definition has been soon neglected by mathematicians for its difficult proof [55], the commonly adopted topological dimension definition is Lebesgue's Covering Dimension, reported in the following.

Definition [Cover] Given a topological space $\boldsymbol{\mathcal{X}}$, a cover of a set $\boldsymbol{\mathcal{Y}} \subseteq \boldsymbol{\mathcal{X}}$ is a countable collection $\boldsymbol{\mathcal{C}} = \{\boldsymbol{\mathcal{C}}_i\}$ of open sets such that each $\boldsymbol{\mathcal{C}}_i \subset \boldsymbol{\mathcal{X}}$ and $\bigcup_i \boldsymbol{\mathcal{C}}_i \supseteq \boldsymbol{\mathcal{Y}}$.

Definition [Refinement of a cover] A refinement of a cover $\boldsymbol{\mathcal{C}}$ of a set $\boldsymbol{\mathcal{Y}}$ is another cover $\boldsymbol{\mathcal{C}}'$ such that each set in $\boldsymbol{\mathcal{C}}'$ is contained in some sets of $\boldsymbol{\mathcal{C}}$.

Definition [Topological Dimension (Lebesgue Covering Dimension)] Given the aforementioned definitions, the Topological Dimension of the topological space $\boldsymbol{\mathcal{X}}$, also called Lebesgue Covering Dimension, is d if every finite cover of $\boldsymbol{\mathcal{X}}$ admits a refinement $\boldsymbol{\mathcal{C}}^{'}$ such that no subset of $\boldsymbol{\mathcal{X}}$ has more than $d+1$ intersecting open sets in $\boldsymbol{\mathcal{C}}^{'}$. If no such minimal integer value exists, $\boldsymbol{\mathcal{X}}$ is said to be of infinite topological dimension.

To our knowledge, at the state-of-the-art only two estimators have been explicitly designed to estimate the topological dimension.

One of them, the Tensor Voting Framework (TVF, [83]) and its variants [78] relies on the usage of an iterative information diffusion process based on Gestalt principles of perceptual organization [125]. TVF iteratively diffuses local information describing, for each $\boldsymbol{p}_i \in \boldsymbol{P}_N$, the tangent space approximating the underlying neighborhood of $\boldsymbol{\mathcal{M}}$. To this aim, the information diffused at each iteration are second order symmetric positive definite tensors whose eigenvectors span the local tangent space. Practically, during the initialization step a ball tensor $\boldsymbol{T}_i^0$, which is an identity matrix representing the absence of orientation, is used to initialize a token p_i for each point $\boldsymbol{p}_i$ as $\{p_i = (\boldsymbol{p}_i, \boldsymbol{T}_i^0)\}_{i=1}^N$. During iteration t each token p_i "generates" the set of tensors $\left\{\boldsymbol{T}_{i,j}^t\right\}_{j \neq i}$ that enact as votes cast to neighboring tokens; precisely, $\boldsymbol{T}_{i,j}^t$ is sent to the j^{th} neighbor, it encodes informations related to the local tangent space estimate in $\boldsymbol{p}_i$ at time t, and decays as the curvature and the distance from the j^{th} neighbor increase. On the other side, at iteration t each token p_j receives votes that are summed to update the p_j's tensor as $\boldsymbol{T}_j^{t+1} = \sum_{i \neq j} \boldsymbol{T}_{i,j}^t$; this essentially refines the estimate of the local tangent space in $\boldsymbol{p}_j$. Based on the definition of topological dimension provided by Brouwer [12], in [86] it is noted that `TVF` can be employed to estimate the local `id`s by identifying the number of most relevant eigenvalues of the computed second order tensors. Although interesting, this method has a too high computational cost, which makes it unfeasible for spaces of dimension $D \geq 4$.

From the definition of Lebesgue Covering Dimension it can be derived [98] that the topological dimension of any $\boldsymbol{\mathcal{M}} \subseteq \Re^d$ coincides with the affine dimension d of a finite simplicial complex[2] covering $\boldsymbol{\mathcal{M}}$. This essentially means that a d-dimensional manifold could be approximated by a collection of d-dimensional simplexes (each having at most $d+1$ vertices); therefore, the topological dimension of $\boldsymbol{\mathcal{M}}$ could be practically estimated by analyzing the number of vertices of the collection of simplexes estimated on $\boldsymbol{P}_N$. To this aim, in [74] a method is proposed to find the number of relevant positive coefficients that are needed to reconstruct each $\boldsymbol{p}_i \in \boldsymbol{P}_N$ from a linear combination of its k neighbors, where k is a parameter to be manually set in the range $d < k \leq D + 1$. This algorithm is based on the fact that neighbors with positive reconstruction coefficients are the vertices of a simplex with dimension equal to the topological dimension of $\boldsymbol{\mathcal{M}}$. Practically, to ensure that $k > d$, its value is set to D, the reconstruction coefficients are calculated by means of an optimization problem constrained to be non negative, and the coefficients bigger than a user-defined threshold are considered as the relevant ones. The `id` estimate is then computed by employing two alternative approaches: the first one simply computes the mode of the number of relevant coefficients for each neighborhood; the second one sorts in descending order the coefficients computed for each neighborhood, computes the mean $\bar{\boldsymbol{c}}$ of the sorted coefficients, and estimates `id` as the number of relevant values in $\bar{\boldsymbol{c}}$. Note that, since $k > d$, this method is strongly affected by the curvature of the manifold when the `id` is big enough. Indeed, the results of the reported experimental evaluation make the authors assert that the method works well only on noisy-free data of low `id` (`id` ≤ 6), under the assumption that the sampling process is uniform and the data points are sufficient.

Though interesting, both this approaches have shown to be effective only for manifolds of low curvature as well as low `id` values.

In the following we survey other `id` estimators employing two different estimation approaches, which allow to categorize them. More precisely: in Section 3.2.1 **fractal `id` estimators** are described, which estimate different fractal dimensions since they are good approximations of the topological one; in Section 3.2.2 **nearest neighbors-based (NN) `id` estimators** are recalled, which are often based on the statistical analysis of the distribution of points within

[2] A simplicial complex in $\Re^d$ has affine dimension d if it is a collection of affine simplexes in $\Re^d$, having at most dimension d, or having at most $d+1$ vertices.

small neighborhoods.

3.2.1 Fractal id Estimators

Since topological dimension cannot be practically estimated, several authors implicitly assume that $\boldsymbol{\mathcal{M}}$ has a somehow fractal structure (see [35] for an exhaustive description of fractal sets) and estimate `id` by employing fractal dimension estimators, the most relevant of which are surveyed in this section.

Very roughly, since the basis concept of all fractal dimensions is that that the volume of a d-dimensional ball of radius r scales with its size s as r^d [35, 114], all fractal dimension estimators are based on the idea of counting the number of observations in a neighborhood of radius r to (somehow) estimate the rate of growth of this number. If the estimated growth is r^d, then the estimated fractal dimension of the data is considered to be equal to d.

Note that all the derived estimators have the fundamental limitation that in order to get an accurate estimation, the size N of the dataset with `id` d has to satisfy the inequality proved by Eckmann in [33] for the correlation dimension estimator (`CD` [45], see below):

$$d < \frac{2}{\log(\frac{1}{\rho})} * \log N, \text{ being } \rho = \frac{r}{D} << 1 \text{ and } \frac{1}{2} N^2 (\frac{r}{D})^d >> 1$$

This will lead to a large value of N, even for a data set with lower `id`.

Among fractal dimension estimators, one of the most cited algorithms is presented in [45] and will be referred as `CD` in the following. It is an estimator of the Correlation Dimension (dim_{Corr}), whose formal definition is:

Definition [Correlation Dimension] Given a finite sample set $\boldsymbol{P}_N$, let:

$$C_N(r) = \frac{2}{N(N-1)} \sum_{i=1, i<j}^{N} I(r - \|\boldsymbol{p_i} - \boldsymbol{p_j}\|) \tag{1}$$

where $\|\cdot\|$ is the Euclidean norm, and $I(\cdot)$ is the step function used to simulate a closed ball of radius r centered on each $\boldsymbol{p_i}$ ($I(y) = 0$ if $y < 0$, and $I(y) = 1$ otherwise). Then, for a countable set, the correlation dimension dim_{Corr} is defined as:

$$dim_{Corr} = \lim_{r \to 0} \lim_{N \to \infty} \frac{log C_N(r)}{log r} \tag{2}$$

In practice `CD` computes an estimate, $\hat{d}$, of dim_{Corr} by computing $C_N(r)$ for different r_i and applying least squares to fit a line through the points ($\log\ r_i$; $\log\ C_N(r_i)$). It has to be noted that, to produce correct `id` estimates, this estimator needs a very large number of data points [119], which is never available for practical applications; however the computed unreliable estimations can be corrected by the correction method proposed in [16].

The relevance of the `CD` estimator is shown by its several variants and extensions. An example is the work proposed in [114], where the authors propose a normalized CD estimator for binary data, and achieve estimates approximating those computed by CD.

Since CD is heavily influenced by the setting of the scale parameters, in [113] the authors estimate the id by computing the expectation value of dim_{Corr} through Maximum Likelihood estimate of the distribution of distances among points. The estimated $\hat{d}$ is computed as:

$$\hat{d} = -\left(\frac{1}{|Q|} \sum_{k=1}^{|Q|} r_k \right)^{-1}$$

where Q is the set $Q = \{r_k | r_k < r\}$, and r_k is the Euclidean distance between two generic data points and r is a real value, called cut-off radius.

To develop an estimator more efficient than CD, in [1] the authors choose a different notion of Fd, namely the Information Dimension dim_I:

$$dim_I = -\lim_{\delta \to 0} \frac{\sum_{i=1}^{\boldsymbol{\mathcal{N}}(\delta)} pr_i (\log pr_i)}{\log \delta}. \tag{3}$$

where $\boldsymbol{\mathcal{N}}(\delta)$ is the minimum number of δ-sized hypercubes covering a topological space and pr_i is the probability of finding a point in the i^{th} hypercube. Noting that, when the scale δ in Equation (3) is big enough the different coverings used to estimate dim_I could produce different values for $\boldsymbol{\mathcal{N}}(\delta)$, the author look for the covering composed by the minimum number $\boldsymbol{\mathcal{N}}_{min}(\delta)$ of nonempty sets. Similar to the CD algorithm, the id is the average slope of the curve obtained by fitting the points with coordinates $\left(\log\ \delta;\ \sum_{i=1}^{\boldsymbol{\mathcal{N}}_{min}(\delta)}\ pr_i \log\ pr_i\right)$.

This algorithm is compared with the CD estimator, and the experimental tests shows that both methods compute the same estimates. However the achieved computation time is much lower than that of CD.

Considering that CD can severely underestimate the topological dimension if the data distribution on the manifold is nearly non-uniform, in [62] the author proposes the Packing Number (PN), a fractal dimension estimator that approximates the Capacity dimension (dim_{Cap}). This choice is motivated by the fact that dim_{Cap} does not depend on the data distribution on the manifold and, if both dim_{Cap} and the topological dimension exist (which is certainly the case in machine learning applications), the two dimensions agree. To formally define dim_{Cap}, the ϵ-covering number $\boldsymbol{\mathcal{N}}(\epsilon)$ of a set $\boldsymbol{\mathcal{S}} \subset \boldsymbol{\mathcal{X}}$ must be defined; $\boldsymbol{\mathcal{N}}(\epsilon)$ is the minimum number of open balls $\boldsymbol{\mathcal{B}}(\boldsymbol{x}_0, \epsilon) = \{\boldsymbol{x} \in \boldsymbol{\mathcal{X}} :\ \|\boldsymbol{x}_0 - \boldsymbol{x}\| < \epsilon\}$ whose union is a covering of $\boldsymbol{\mathcal{S}}$, where $\|\cdot\|$ is a distance metric. The definition of dim_{Cap} of $\boldsymbol{\mathcal{S}} \subset \boldsymbol{\mathcal{X}}$ is based on the observation that the covering number $\boldsymbol{\mathcal{N}}(\epsilon)$ of a d-dimensional set is proportional to ϵ^{-d}:

$$dim_{Cap} = -\lim_{\epsilon \to 0} \frac{\log \boldsymbol{\mathcal{N}}(\epsilon)}{\log \epsilon}. \tag{4}$$

Since the estimation of $\boldsymbol{\mathcal{N}}(\epsilon)$ is computationally expensive, based on the relation $\boldsymbol{\mathcal{N}}(\epsilon) \leq \boldsymbol{\mathcal{N}}_{Pack}(\epsilon) \leq \boldsymbol{\mathcal{N}}(\frac{\epsilon}{2})$, the authors employ the ϵ-Packing number

$\boldsymbol{\mathcal{N}}_{Pack}(\epsilon)$, defined in [117] as the maximum cardinality of an ϵ-separated set. Employing a greedy algorithm to compute $\boldsymbol{\mathcal{N}}_{Pack}(\epsilon)$, the estimate, $\hat{d}$, of dim_{Cap} is computed as:

$$\hat{d}\,(\epsilon_1, \epsilon_2) = -\frac{\log \boldsymbol{\mathcal{N}}_{Pack}(\epsilon_1) - \log \boldsymbol{\mathcal{N}}_{Pack}(\epsilon_2)}{\log \epsilon_1 - \log \epsilon_2} \tag{5}$$

To estimate $\hat{d}$ a greedy algorithm is used; however, as noted by the author, the dependency of $\hat{d}$ w.r.t. the order in which the points are visited by the greedy algorithm introduces a high variance. To avoid this problem, the algorithm iterates the procedure M times on random permutations of the data, and considers the average as the final `id` estimate. The comparative evaluation with the `CD` estimator make the authors assert that `PN` "seems more reliable if data contains noise or the distribution on the manifold is not uniform". Unfortunately, also this method is scale-dependent.

To avoid any scale-dependency in [49] the authors propose an estimator (`Hein`) based on the asymptotes of a smoothed version of Equation (1), obtained by replacing the step function $I(\cdot)$ with a suitable kernel function. Precisely, they define:

$$U(N, h, d) = \frac{2}{N(N-1)} \sum_{1 \le i < j \le N}^{N} \frac{1}{h^d} K_h(\|\boldsymbol{p}_i - \boldsymbol{p}_j\|/h^2). \tag{6}$$

where K_h is a kernel function with bandwidth h, and d is the assumed dimensionality of the manifold from which the points are sampled. Note that, to guarantee the converge of Equation (6), the bandwidth h has to fulfill the constraint $\lim_{N\to\infty}(Nh^d) = \infty$. For this reason the authors formalize h as a function of N and, to achieve scale-independency, propose a method that estimates the `id` by analyzing the convergence of $U(N, h, d)$ when varying the parameters N and d. Precisely, the dataset is sub-sampled to create sets of different cardinalities $n_{sub} \in \boldsymbol{\mathcal{N}}_{sub} = \{N, N/2, N/3, N/4, N/5\}$ and the D curves whose points have coordinates $(U(n_{sub}, h(n_{sub}), d), n_{sub})$ are considered. Employing this information the following `id` estimator is proposed:

$$\begin{aligned} Slope(d) &= \max_{n_{sub} \in \boldsymbol{\mathcal{N}}_{sub}} \left| \frac{\partial U(n_{sub}, h(n_{sub}), d)}{\partial n} \right| \\ \hat{d} &= \underset{d \in \{1..D\}}{\arg\min}\, Slope(d) \end{aligned}$$

This work is notable since the empirical tests are performed on synthetic datasets specifically designed to study the influence of high curvature as well as noise on the proposed estimator. The usefulness of these datasets is confirmed by the fact that they have been also employed to assess several subsequent methods [11, 20].

In [96] the authors present a fractal dimension estimator derived by the analysis of a vector quantizer applied to datasets $\boldsymbol{P}_N \subseteq \Re^D$. Considering the codebook $\boldsymbol{\mathcal{Y}} = \{\boldsymbol{y}_1..\boldsymbol{y}_k\} \subset \Re^D$ containing k code-vectors $\boldsymbol{y}_i$, a *k-point quantizer* is defined by a measurable function $Q_k : \Re^D \to \boldsymbol{\mathcal{Y}}$, which brings each data point to one of the code-vectors in $\boldsymbol{\mathcal{Y}}$. This partitions the dataset into k so-called *quantizer cells* $\boldsymbol{\mathcal{S}}_i = \{\boldsymbol{p}_i \in \boldsymbol{P}_N : Q_k(\boldsymbol{p}_i) = \boldsymbol{y}_i\}$, where $\log_2(k)$ is called the *rate of the quantizer*. Being $\boldsymbol{X}$ a random vector distributed according to a probability distribution ν, the *quantization error* is $e_r(Q_k|\nu) = (E_\nu[\|\boldsymbol{X} - Q_k(\boldsymbol{X})\|^r])^{\frac{1}{r}}$, where $r \in [1, \infty)$ and $\|\cdot\|$ is the Euclidean norm in $\Re^D$. Given the set $\boldsymbol{\mathcal{Q}}_k$ of all D-dimensional k-point quantizers, the performance achieved by an optimal k-point quantizer on $\boldsymbol{X}$, is $e^*_r(Q_k|\nu) = \inf_{Q_k \in \boldsymbol{\mathcal{Q}}_k}(e_r(Q_k|\nu))$. When the quantizer rate is high, the quantizer cells can be well approximated by D-dimensional hyper-spheres with radius equal to ϵ and centered on each code-vector $\boldsymbol{y}_i \in \boldsymbol{\mathcal{Y}}$. In this case, the regularity of ν ensures that the probability of such balls is proportional to $\epsilon^{\frac{1}{d}}$, and it can be shown [126] that $e^*_r(Q_k|\nu) \approx k^{-\frac{1}{d}}$. This is referred to as the *high-rate approximation*, and motivates the definition of Quantization Dimension of order r:

$$d_r(\nu) = -\lim_{k \to \infty} \frac{\log k}{\log e^*_r(k|\nu)}$$

The theory of high-rate quantization [126] confirms that, for a regular ν supported on the manifold $\boldsymbol{\mathcal{M}}$, $d_r(\nu)$ exists for each $1 \leq r \leq \infty$ and equals the intrinsic dimension of $\boldsymbol{\mathcal{M}}$. Furthermore, the limit $k \to \infty$ allows to motivate the relation between the quantization dimension and the Capacity Dimension. Indeed, according to the theory of high-rate quantization [126, 57], there exists a decreasing sequence $\{\epsilon_k\}$, such that for sufficiently large values of k (i.e., in the high-rate regime that is when $k \to \infty$) the ratio $-\frac{\log k}{\log e^*_r(k|\nu)}$ can be approximated increasingly finely, both from below and from above, by quantities converging to the common value dim_{Cap}. To practically compute an estimate of the quantization dimension, having fixed the value of r, the authors select a range $k_1 \leq k \leq k_2$ of codebook sizes, and design a set of quantizers $\{Q_k\}_{k=k_1}^{k_2}$ giving good approximations $\hat{e}_r(k|\nu)$ of $e^*_r(k|p)$ over the chosen range of k. An `id` estimate is obtained by fitting the points with coordinates $(\log(k); -\log \hat{e}_r(k|\nu))$ and measuring the average slope over the chosen range k. Though the authors mention that their algorithm is less affected by underestimation biases than neighborhood-based methods (see Section 3.2.2), in [18] this statement is confuted with theoretical arguments.

3.2.2 Nearest Neighbors-based `id` Estimators

In this section we consider estimators, referred as NN estimators in the following, that describe data-neighborhoods' distributions as functions of d. They usually assume that close points are uniformly drawn from small d-dimensional balls (hyperspheres) $\boldsymbol{\mathcal{B}}_d(\boldsymbol{x}, r)$ having radius[3] $r \to 0 \in \Re^+$ and being centered on $\boldsymbol{x} \in \boldsymbol{\mathcal{M}}$.

[3] A small radius $r \to 0 \in \Re^+$ guarantees that samples included into $\boldsymbol{\mathcal{B}}_d(\boldsymbol{x}, r)$, being less influenced by the curvature induced by the map ϕ, are approximating well enough the intrinsic structure of the underlying portion of $\boldsymbol{\mathcal{M}}$.

Practically, given an input dataset $\boldsymbol{P}_N$, the value of functions $f(d)$ is computed by approximating the sampling process related to $\boldsymbol{\mathcal{B}}_d$ through the k-Nearest Neighbor algorithm (`kNN`).

Among NN `id` estimators, Trunk's method [118] is often cited as one of the first. It formulates the distribution function, $f(d)$, with an ad-hoc statistic based on geometric considerations concerning angles; in practice, having fixed a threshold γ and a starting value for the parameter k, it applies `kNN` to find the neighbors of each $\boldsymbol{p}_i \in \boldsymbol{P}_N$, and calculates the angle ν_i between the $(k+1)^{th}$-nearest neighbor and the subspace spanned by the k-nearest neighbors. Considering a threshold parameter γ, if $\frac{1}{N}\sum_{i=1}^{N}\nu_i \leq \gamma$, then k is considered as the `id` estimate, otherwise k is incremented by 1 and the process is repeated. The main limitation of this method is the difficult choice of a proper value for the γ.

The work presented by Pettis [92] is notable since it is one of the first providing a mathematical motivation for the use of nearest-neighbor distances.

Indeed, for an i.i.d. sample $\boldsymbol{P}_N \subseteq \Re^D$ drawn from a density distribution $p(\boldsymbol{x})$ in $\Re^d$, the following approximation holds:

$$\frac{k}{N} \simeq p(\boldsymbol{x})V(d)r^d \tag{7}$$

where k is the number of nearest neighbors to $\boldsymbol{x}$ within the hypersphere $\boldsymbol{\mathcal{B}}_d(\boldsymbol{x}, r)$ of radius r and centered on $\boldsymbol{x}$, and $V(d)$ is the volume of the (unit d-dimensional) ball in $\Re^d$.

This means that the proportion of sample points falling in $\boldsymbol{\mathcal{B}}_d(\boldsymbol{x}, r)$ is roughly approximated by $p(\boldsymbol{x})$ times the volume of $\boldsymbol{\mathcal{B}}_d(\boldsymbol{x}, r)$. Since this volume grows as r^d, assuming the density $p(x)$ to be a constant, it follows that the number of samples in $\boldsymbol{\mathcal{B}}_d(\boldsymbol{x}, r)$ is proportional to r^d. From the relationship in Equation (7), and assuming that the samples are locally uniformly distributed, the authors derive an `id` estimator for d:

$$\hat{d} = \frac{\bar{r}_k}{k(\bar{r}_{k+1} - \bar{r}_k)}$$

where $\bar{r}_k$ is the average of the distances from each sample point to its k^{th} nearest neighbors; defining $r_i^{(k)}$ as the distance between $\boldsymbol{x}_i$ and its k^{th}-nearest neighbor, $\bar{r}_k$ is expressed as $\bar{r}_k = \frac{1}{N}\sum_{i=1}^{N} r_i^{(k)}$.

Since this algorithm is limited by the choice of a suitable value for parameter k, in [122] the authors propose a variant which considers a range of neighborhood sizes $[k_{min}, k_{max}]$. However, in the same work the authors themselves show that this technique generally yields an underestimate of the `id` when its value is high.

Taking into account the relation Equation (7), in [36] the number $N_{\boldsymbol{\mathcal{B}}_d}$ of data points in $\boldsymbol{\mathcal{B}}_d(\boldsymbol{x}, r)$ is described by a polynomial $f(r) = \sum_{s=0}^{d}\beta_s r^s$ of degree d. In practice, considering $\boldsymbol{p}_i, \boldsymbol{p}_k \in \boldsymbol{P}_N$, calling $r_{ik} = \|\boldsymbol{p}_i - \boldsymbol{p}_k\|$ the inter-point distances, and being $r = \min_{i,k=1}^{N} r_{ik}$, and R a parameter adaptively estimated[4],

[4] To estimate R by means of $\boldsymbol{P}_N$, the radius value corresponding to the first significant peak of the histogram of the r_{ij}s is found.

a set of n radius values $\boldsymbol{r} = \left\{ r_j = r + \frac{j(R-r)}{n} \right\}_{j=1}^{n}$ is selected and used to calculate n pairs $\left\{ \left(r_j, \hat{f}(r_j) \right) \right\}_{j=1}^{n}$, where $\hat{f}(r_j) = \# \left[r_{ik} < r_j \right]_{i,k=1}^{N}$ is the number of inter-point distances strictly lower than r_j. To estimate the coefficients $\{\beta_j\}_{j=1}^{D}$, the computed pairs are fit by a least squares fitting procedure that estimates exactly $D+1$ coefficients. Since by hypothesis the degree of f is d, the significance test described in [40] is used to estimate the degree $\hat{d}$ of $\hat{f}$, which is considered as the `id` estimate. The comparative evaluation of this algorithm with the well-known Maximum Likelihood Estimator (`MLE` [73]) and its improved version [80], both described below, has shown that it is more robust than them when dealing with high dimensional datasets.

`MLE` [73], one of the most cited estimators, treats the neighbors of each point $\boldsymbol{p}_i \in \boldsymbol{P}_N$ as events in a Poisson process and the distance $r^{(j)}(\boldsymbol{p}_i)$ between the query point $\boldsymbol{p}_i$ and its j^{th} nearest neighbor as the event's arrival time. Since this process depends on d, `MLE` estimates `id` by maximizing the log-likelihood of the observed process. In practice a local `id` estimate is computed as:

$$\hat{d}(\boldsymbol{p}_i, k) = \left(\frac{1}{k} \sum_{j=1}^{k} \log \frac{r^{(k+1)}(\boldsymbol{p}_i)}{r^{(j)}(\boldsymbol{p}_i)} \right)^{-1}$$

Averaging the $\hat{d}(\boldsymbol{p}_i, k)$s, the global `id` estimate is $\hat{d}(k) = \frac{1}{N} \sum_{i=1}^{N} \hat{d}(\boldsymbol{p}_i, k)$.

The theoretical stability of the proposed `id` estimator for data living in C^1 submanifold of $\Re^D$, $d \leq D$, and for data in an affine subspace of $\Re^D$ has been proved respectively in [89, 5]. Though the authors' comparative evaluation shows the superior performance of the proposed estimator w.r.t. the `CD` estimator [45] (see Section 3.2.1) and the `NN` estimator [92], they further improve it by removing its dependency from the parameter k; to this end, different values for k are adopted and the computed results are averaged to obtain the final `id` estimate: $\hat{d} = \frac{1}{t} \sum_{k \in \{k_1..k_t\}} \hat{d}(k)$.

Considering that, in practice, `MLE` is highly biased both for large and small values of k, a variant of `MLE` is proposed in [80], where the arithmetic mean is substituted with the harmonic average, leading to the following estimator: $\hat{d}(k) = \left(\frac{1}{N} \sum_{i=1}^{N} \frac{1}{\hat{d}(\boldsymbol{p}_i, k)} \right)^{-1}$.

Though the proposal in [80] seems to achieve more accurate results, it is based on the assumption that neighbors surrounding each $\boldsymbol{p}_i$ are independent, which is clearly incorrect. To cope with this problem, in [30] an interesting regularized version of `MLE` applies a regularized maximum likelihood technique to distances between neighbors. The comparative evaluation with the aforementioned `MLE` methods [73, 80] make the authors state that, though the method might be the first to converge to the actual estimate given enough data points, its estimation accuracy is comparable to that achieved by the competing schemes.

In [59, 58] a further improvement of `MLE` is presented; it achieves a better performance by substituting euclidean distances with geodesic ones.

Despite the good results achieved by MLE-based approaches, these techniques have shown to be affected by the curvature induced by ϕ on the manifold neighborhoods approximated by kNN. To reduce this effect, various id estimators have been proposed in [102, 20]; here, we review those achieving the most promising experimental results.

In [102] the authors firstly propose a family of id estimators ($\mathtt{MiND}_{\mathtt{ML*}}$), which exploit the pdf $g(r;k,d)$ describing the distance $r^{(1)}(\boldsymbol{x})$ between the center $\boldsymbol{x}$ of $\boldsymbol{\mathcal{B}}_d(\boldsymbol{x},r)$, $\boldsymbol{x} \in \boldsymbol{\mathcal{M}}$, $r \to 0^+$ and its nearest neighbor. Briefly, formulating $g(r;k,d)$ as a function of the id value d ($g(r;k,d) = kdr^{d-1}(1-r^d)^{k-1}$), the id estimator is computed by a maximum likelihood approach.

After noting that this algorithm is still affected by a bias causing underestimations when the dataset dimensionality becomes sufficiently high (that is $id \geq 10$), the authors present theoretical considerations which relate the bias to the fact that id estimators based on nearest-neighbor distances are often founded on statistics derived under the assumption that the amount of available data is unlimited, which is never the case in practical applications. Based on this considerations, two different estimators, named $\mathtt{MiND}_{\mathtt{KL}}$ and IDEA are presented.

$\mathtt{MiND}_{\mathtt{KL}}$ compares the empirical pdf of the neighborhood distances computed on the dataset (g_{Data}) with the distribution of the neighborhood distances computed from points uniformly drawn from hyperspheres of known increasing dimensionality (g^d_{Sphere}). The id estimate is the dimensionality that minimizes their Kullback-Leibler divergence $\boldsymbol{\mathcal{KL}}(g_{Data}, g^d_{Sphere})$, which is evaluated by means of the data-driven technique proposed in [123].

IDEA relies on the authors' observation that the quantities $1 - \frac{r^{(j)}(\boldsymbol{p}_i)}{r^{(k+1)}(\boldsymbol{p}_i)}$ are distributed according to the beta distribution $\beta_{1,d}$ with parameters 1 and d respectively. Therefore, since $\mathbb{E}\left[\beta_{1,d}\right] = m = \frac{1}{1+d}$, a consistent id estimator $\hat{d} \simeq d$ equals:

$$\hat{d} = \frac{\hat{m}}{1-\hat{m}} \simeq d = \frac{m}{1-m} \quad \text{where} \quad \hat{m} = \frac{1}{Nk}\sum_{i=1}^{N}\sum_{j=1}^{k}\frac{r^{(j)}(\boldsymbol{p}_i)}{r^{(k+1)}(\boldsymbol{p}_i)} \simeq m$$

To reduce the effect of the aforementioned bias, IDEA finally applies an asymptotic correction step that, inspired to the correction method presented in [16], models the underestimation error by considering both the base algorithm and the given dataset.

Motivated by the promising results achieved by $\mathtt{MiND}_{\mathtt{KL}}$, in [20] the authors propose its extension, namely DANCo; it further reduces the underestimation effect by combining an estimator employing normalized nearest-neighbor distances with one employing mutual angles. More precisely, DANCo compares the statistics estimated on $\boldsymbol{P}_N$ with those estimated on (uniformly drawn) synthetic datasets of known id. The comparisons are performed by two Kullback-Leibler divergences applied to the distribution of normalized nearest-neighbor distances $g(r;k,d)$, being $g(r;k,d) = kdr^{d-1}(1-r^d)^{k-1}$, and the distribution of pairwise angles $q(\boldsymbol{x};\boldsymbol{\nu},\tau)$, being $q(\boldsymbol{x};\boldsymbol{\nu},\tau)$ the von Mises-Fisher distribution (`VMF`, [81]) with parameters $\boldsymbol{\nu}$ and τ.

The `id` estimate $\hat{d}$ is the one minimizing the sum of the two divergences:

$$\hat{d} = \underset{d\in\{1..D\}}{\arg\min}\, \boldsymbol{\mathcal{KL}}(g_{Data}, g^{d}_{Sphere}) + \boldsymbol{\mathcal{KL}}(q_{Data}, q^{d}_{Sphere}).$$

A fast implementation of `DANCo` (Fast-`DANCo`) is also developed. Comparative evaluations show that this algorithm achieves promising results (as shown in [20] and Section 4).

Another work, which is notable because the authors not only prove the consistency in probability of the presented estimators, but they also derive upper bounds (see Equation (8) below) on the probability of the estimation-error for finite, and large enough, values of N, is proposed in [37]. More precisely, the authors introduce two estimators by firstly defining a function $\eta : \Re^D \times \Re \to \Re^+$ slowly varying near the origin(see [37] for a detailed description and motivation of this assumption). The function η is then used to express the logarithm of the probability of a point $\boldsymbol{p}$ of being in the hypersphere $\boldsymbol{\mathcal{B}}_D(\boldsymbol{p}_i, r)$: $\log\left(\mathbf{P}\left(\boldsymbol{p} \in \boldsymbol{\mathcal{B}}_D(\boldsymbol{p}_i, r)\right)\right) = \log\left(\eta(\boldsymbol{p}, r)\right) + d\log(r)$, being $\mathbf{P}\left(\boldsymbol{p} \in \boldsymbol{\mathcal{B}}_D(\boldsymbol{p}_i, r)\right) = \eta(\boldsymbol{p}, r) r^d$.

Considering that $\mathbf{P}\left(\boldsymbol{p} \in \boldsymbol{\mathcal{B}}(\,\boldsymbol{p}_i, r^{(k)}(\boldsymbol{p}_i)\,)\right) \approx {}^{k}\!/_{n}$ for N big enough, the authors derive the following system of equations:

$$\begin{aligned} \log(k/n) &\approx \log\left(\eta(\boldsymbol{p}_i, r)\right) + \hat{d}(\boldsymbol{p}_i)\log(r^{(k)}(\boldsymbol{p}_i)) \\ \log(k/(2n)) &\approx \log\left(\eta(\boldsymbol{p}_i, r)\right) + \hat{d}(\boldsymbol{p}_i)\log(r^{(\lceil k/2\rceil)}(\boldsymbol{p}_i)) \end{aligned}$$

and solve it for $\hat{d}(\boldsymbol{p}_i)$ to obtain a local `id` estimate:

$$\hat{d}(\boldsymbol{p}_i) = \frac{\log(2)}{\log\left({}^{r^{(k)}(\boldsymbol{p}_i)}\!/_{r^{(\lceil k/2\rceil)}(\boldsymbol{p}_i)}\right)}.$$

The two proposed estimators are then computed either by averaging ($\hat{d}_{avg}$) or by voting ($\hat{d}_{vote}$):

$$\begin{aligned} \hat{d}_{avg} &= \frac{1}{N}\sum_{i=1}^{N}\hat{d}(\boldsymbol{p}_i) \\ \hat{d}_{vote} &= \underset{d'\in\mathbb{N}^+}{\arg\max}\, \#\left[\hat{d}(\boldsymbol{p}_i) = d'\right]_{i=1}^{N}, \end{aligned}$$

where $\#\left[cond\right]_{i=1}^{N}$ denotes the number of points $\boldsymbol{p}_i$ for which *cond* is verified.

Under differentiability assumptions on the function η and regularity assumptions on $\boldsymbol{\mathcal{M}}$ the authors prove the consistency in probability of their estimators and provide upper bounds (see Equation (8)) on the probability of the estimation-error for finite, and large enough, values of N. However, the derived bounds depend on unknown universal constants $c, c', c'' > 0$.

$$\mathbf{P}\left(\hat{d}_{avg} \neq d\right) \leq \exp\left(-\frac{c' N}{(Dc^d k)^2}\right) \qquad \mathbf{P}\left(\hat{d}_{vote} \neq d\right) \leq \exp\left(-\frac{c'' N}{(c^d k)^2}\right) \quad (8)$$

3.3 Graph-based `id` Estimators

As noted in [11], the work of [95] has cleared that theories underlying graphs can be applied to solve a variety of statistical problems; indeed, also in the field of `id` estimation various types of graph structures have been proposed [11, 50, 27, 26] and used for `id` estimation. Examples are the `kNN` graph (`kNNG`, [28]), the Minimum Spanning Tree (`MST`, [39]) and its variation, the geodesic `MST` (`GMST`, [27]), the sphere of influence graph (`SIG`, [88]), and its generalization, the $k-$sphere of influence graph (`kSIG`, [11]).

Given a sample set $\boldsymbol{P}_N = \{\boldsymbol{p}_i\}_{i=1}^N$ a graph built on $\boldsymbol{P}_N$, usually denoted with $G(\boldsymbol{P}_N) = (\{\boldsymbol{p}_i\}_{i=1}^N, \{e_{i,j}\}_{i,j\in\{1..N\}})$, employs the sample points $\boldsymbol{p}_i$ as nodes (vertices) of the graph, and connects them with weighted arcs (edges) $\{e_{i,j}\}_{i,j\in\{1..N\}}$.

A $\texttt{kNNG}_N(\boldsymbol{P}_N)$ is built by employing a distance function, which commonly is the Euclidean one, to weight the arcs connecting each $\boldsymbol{p}_i$ to its `kNN`s.

A $\texttt{MST}(\boldsymbol{P}_N)$ is the spanning tree minimizing the sum of the edge weights. When the weights approximate Geodesic distances [115], a $\texttt{GMST}_N(\boldsymbol{P}_N)$ is obtained.

A $\texttt{SIG}_N(\boldsymbol{P}_N)$ is defined by connecting nodes $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$ iff $||\boldsymbol{p}_i - \boldsymbol{p}_j|| \leq \rho(i) + \rho(j)$, where $\rho(i)$ is the distance between $\boldsymbol{p}_i$ and its nearest neighbor in $\boldsymbol{P}_N$. Essentially, two vertices $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$ are connected if the corresponding `NN` hyperspheres intersect. A generalization of $\texttt{SIG}_N(\boldsymbol{P}_N)$ is $\texttt{kSIG}(\boldsymbol{P}_N)$, where nodes $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$ are connected iff $||\boldsymbol{p}_i - \boldsymbol{p}_j|| \leq \rho_k(i) + \rho_k(j)$, being $\rho_k(i)$ the distance between $\boldsymbol{p}_i$ and its `kNN` in $\boldsymbol{P}_N$. This means that the `kNN` hyperspheres centered on $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$ intersect.

In the following we recall interesting `id` estimators based on $\texttt{GMST}(\boldsymbol{P}_N)$, $\texttt{kNNG}(\boldsymbol{P}_N)$, and $\texttt{kSIG}(\boldsymbol{P}_N)$.

In [27, 28], after defining the length functional $\boldsymbol{\mathcal{L}}(G_N(\boldsymbol{P}_N)) = \sum |e_{i,j}|^\gamma$, $\gamma \in (0, d)$, to build either the $\texttt{GMST}(\boldsymbol{P}_N)$ or the $\texttt{MST}(\boldsymbol{P}_N)$ of $\texttt{kNNG}(\boldsymbol{P}_N)$, graph theories are exploited to estimate both the `id` of the underlying manifold structure $\boldsymbol{\mathcal{M}}$ and its intrinsic Rènyi α-entropy $\boldsymbol{\mathcal{H}}_{\boldsymbol{\mathcal{M}}}$. To this aim, the authors derive the linear model: $\log \boldsymbol{\mathcal{L}}(\texttt{MST}(\boldsymbol{P}_N)) = a \log d + b,\ a = (d-\gamma)/d,\ b = \log c + \boldsymbol{\mathcal{H}}_{\boldsymbol{\mathcal{M}}}$, being c an unknown constant, and exploit it to define an estimator of both d and $\boldsymbol{\mathcal{H}}_{\boldsymbol{\mathcal{M}}}$. Briefly, a set of cardinalities $\{n_k\}_{k=1}^K$ is chosen and, for each n_k, the $\texttt{MST}(\boldsymbol{P}_{n_k})$ is constructed on the set $\boldsymbol{P}_{n_k}$, which contains n_k points randomly sampled from $\boldsymbol{P}_N$, to obtain a set of K pairs $(\log \boldsymbol{\mathcal{L}}\left(\texttt{MST}(\boldsymbol{P}_{n_k})\right), n_k)$. Fitting them with a least squares procedure the estimates $\hat{a} \simeq a$ and $\hat{b} \simeq b$ are computed. Recalling that $a = (d-\gamma)/d$, the `id` is calculated as $\hat{d} = \text{round}\big\{\gamma/(1-\hat{a})\big\} \simeq d$. This process is iterated to produce the final estimate as the average of the obtained results.

The aforementioned `kNNG` based algorithm [28, 27] is exploited in [26], where the authors consider data sets sampled from a union of disjoint manifolds with possibly different `id`s. To estimate the local `id`s, the authors propose an heuristic, which is not described here, to automatically determine the local neighborhoods with similar geometric structures without any prior knowledge on the number of manifolds, their `id`s, and their sampling distributions.

In [11] the authors present three `id` estimation approaches, defined as “graph theoretic methods” since the statistics they compute are functions only of graph properties (such as vertex degrees, vertex eccentricities, and so on) and do not directly depend from the inter-point distances.

The first statistic, denoted as $S^1_N(\boldsymbol{P}_N) = \bar{r}_j(\texttt{kNNG}(\boldsymbol{P}_N))$ in the following, is based on the reach[5] of vertices in the $\texttt{kNNG}(\boldsymbol{P}_N)$. Considering that the reach of each vertex $\boldsymbol{p}_i \in \texttt{kNNG}(\boldsymbol{P}_N)$ grows as the `id` increases, in [10] the average reach $\bar{r}_j(\texttt{kNNG})$ in j steps of vertices in $\texttt{kNNG}(\boldsymbol{P}_N)$ is employed: $S^1_N(\boldsymbol{P}_N) = \bar{r}_j(\texttt{kNNG}(\boldsymbol{P}_N)) = \frac{1}{N}\sum_{i=1}^{N} r_{j,i}(\boldsymbol{p}_i, \texttt{kNNG}(\boldsymbol{P}_N))$.

The second statistic, denoted with $S^2_N(\boldsymbol{P}_N) = M_N(\texttt{MST}(\boldsymbol{P}_N))$, is computed by considering the degree of vertices in the $\texttt{MST}(\boldsymbol{P}_N)$. Recalling that, for datasets $\boldsymbol{P}_N$ obtained from a continuous distribution on $\Re^d$, the ratio of nodes with a given degree j in $\texttt{MST}_N(\boldsymbol{P}_N)$ converges a.s. to a limit depending only on j and d [112], and that the average degree in a tree is a constant depending only on the number of vertices, the authors empirically observe a dependency between the average degree and the `id`. This leads to the definition of an `id` estimator employing the statistic $S^2_N = M_N(\texttt{MST}(\boldsymbol{P}_N)) = \frac{1}{N}\sum_{i=1}^{N}(deg_{MST(\boldsymbol{P}_N)}(\boldsymbol{p}_i))^2$.

The third statistic, denoted as $S^3_N(\boldsymbol{P}_N) = U^k_N(\texttt{kSIG}(\boldsymbol{P}_N))$, is motivated by studies in the literature [109] showing that the expected number of neighbors shared by a given pair of points depends on the `id` of the underlying manifold. Accordingly, calling $N_{i,j}$ the number of samples in the intersection of the two `kNN` hyperspheres centered on $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$, intuitions similar to those considered for $\bar{r}_j(\texttt{kNNG})$ lead to define $S^3_N(\boldsymbol{P}_N) = U^k_N(\texttt{kSIG}(\boldsymbol{P}_N)) = \frac{1}{n}\sum_{i\leq j} N_{i,j}$.

Based on their theoretical results and empirical tests on synthetically generated datasets characterized by `id` values d_j in a finite range $\boldsymbol{F} \subseteq N^+$ (where $\boldsymbol{F} = \{d_j\}_{d_j=2}^{12}$ in the reported experiments), the authors propose an approximate Bayesian estimator that could indistinctly employ each of the three statistics S^1_N, S^2_N, and S^3_N, denoted by S^*_N in the following. To this aim, they assume that each statistic can be approximated by a Gaussian density $f_{d_j}(\cdot) = \mathcal{N}(\mu(d_j), \sigma^2(d_j))$; to estimate $\mu(d_j)$ and $\sigma^2(d_j)$, for each $d_j \in \boldsymbol{F}$, L datasets of large size are synthetically generated by random sampling from the Uniform distribution on the unit d_j-cube. These datasets are then used to estimate the parameters $\tilde{\mu}(d_j) \simeq \mu(d_j)$ and $\tilde{\sigma}^2(d_j) \simeq \sigma^2(d_j)$ that define the approximation $\tilde{f}_{d_j}(\cdot)$, computed on a generic sample set with size N and $\texttt{id} = d_j$, of the Gaussian density $f_{d_j}(\cdot)$ of S^*_N.

At this stage, given a new input dataset $\boldsymbol{P}_N$ having unknown `id`, the statistic $S^*_N(\boldsymbol{P}_N) = \boldsymbol{s}_{\boldsymbol{P}}$ is computed and used to calculate the approximated value $\tilde{f}_{d_j}(s_{\boldsymbol{P}}) = \mathcal{N}(\tilde{\mu}^2(d_j), \frac{\tilde{\sigma}^2(d_j)}{N}) \simeq f_{d_j}(s_{\boldsymbol{P}})$. Assuming equal a priori probability for all the $d_j \in \boldsymbol{F}$, the posterior probability $P[d_j|S^*_N]$ is given by:

$$P[d_j|S^*_N] = \frac{\tilde{f}_{d_j}(s_{\boldsymbol{P}})}{\sum_{d_j\in F}\tilde{f}_{d_j}(s_{\boldsymbol{P}})},\ d_j \in \boldsymbol{F}$$

[5] The reach $r_{j,i}(\boldsymbol{p}_i, G)$, in j steps of a node $\boldsymbol{p}_i \in G$, is the total number of vertices which are connected to $\boldsymbol{p}_i$ by a path composed of j arcs or less in G.

and employed to compute an "a posteriori expected value" of the id:

$$\hat{d} = \mathrm{round}\Big\{ \sum_{d_j \in F} d_j P[d_j | S_N^*] \Big\}.$$

The authors evaluate the performance of their methods on synthetic datasets, some of which have been used by similar studies in literature [49], while the others (challenging ones) are proposed by the authors to have manifolds with non-constant curvature. The comparison of the achieved results with those obtained by the estimators proposed in [73, 37, 26, 111] has lead to the conclusion that none of the methods has a good performance on all the tested datasets. However, graph theoretic approaches would appear to behave better when manifolds of non-constant curvature are processed.

This interesting comparison strengthen the need of defining a benchmark framework to allow an objective and reproducible comparative evaluation of id estimators. For this reason, in Section 4 we describe our proposal in this direction.

4 A Benchmark Proposal

At the present, an objective comparison of different id estimators is not possible due to the lack of a standardized benchmark framework; therefore in this section, after recalling experimental datasets and evaluation procedures introduced in literature (see Sections 4.1, 4.2), we choose some of them to propose a benchmark framework (see Section 4.3) that allows for reproducible and comparable experimental setups. The usefulness of the proposed benchmark is then shown by employing it to compare relevant state-of-the-art id estimators whose code is publicly available (see Section 4.4).

4.1 Datasets

The datasets employed in literature are both synthetically generated datasets and real ones. In the following sections we describe those we choose to use in our benchmark study.

4.1.1 Synthetic Datasets

Synthetic datasets are generated by drawing samples from manifolds ($\boldsymbol{\mathcal{M}}$) linearly or non-linearly embedded in higher dimensional spaces.

The publicly available tool[6] proposed by Hein in [49] allows to generate 13 kinds of synthetic datasets by uniformly drawing samples from 13 manifolds of known id; they are schematically described in Table 1, where they are referred to as $\boldsymbol{\mathcal{M}}_*^H$. These manifolds are embedded in higher dimensional spaces through both linear and non-linear maps and are characterized by different curvatures.

[6] http://www.ml.uni-saarland.de/code/IntDim/IntDim.htm

Dataset	**Underlying Manifold Name**	**Description**	$\boldsymbol{d}$	$\boldsymbol{D}$
Syntethic	$\boldsymbol{\mathcal{M}}_1^H$	d-dimensional sphere linearly embedded.	$\boldsymbol{D}-1$	*User Defined*
	$\boldsymbol{\mathcal{M}}_2^H$	Affine space.	3	5
	$\boldsymbol{\mathcal{M}}_3^H$	Concentrated figure, mistakable with a 3-dimensional one.	4	6
	$\boldsymbol{\mathcal{M}}_4^H$	Non-linear manifold.	4	8
	$\boldsymbol{\mathcal{M}}_5^H$	2-dimensional Helix	2	3
	$\boldsymbol{\mathcal{M}}_6^H$	Non-linear manifold.	6	36
	$\boldsymbol{\mathcal{M}}_7^H$	Swiss-Roll.	2	3
	$\boldsymbol{\mathcal{M}}_8^H$	Non-linear (highly curved) manifold.	12	72
	$\boldsymbol{\mathcal{M}}_9^H$	Affine space.	$\boldsymbol{D}$	*User Defined*
	$\boldsymbol{\mathcal{M}}_{10}^H$	d-dimensional hypercube.	$\boldsymbol{D}-1$	*User Defined*
	$\boldsymbol{\mathcal{M}}_{11}^H$	Möebius band 10-times twisted.	2	3
	$\boldsymbol{\mathcal{M}}_{12}^H$	Isotropic multivariate Gaussian.	$\boldsymbol{D}$	*User Defined*
	$\boldsymbol{\mathcal{M}}_{13}^H$	1-dimensional Helix Curve.	1	*User Defined*

Table 1: The 13 types of synthetic datasets generated with the tool proposed in [49].

We note that manifold $\boldsymbol{\mathcal{M}}_8^H$ is particularly challenging for its high curvature; indeed, when it is used for testing, most relevant `id` estimators compute pronounced `id` overestimates (see also the results reported in [102]).

Another interesting dataset [11] is generated by sampling a d-dimensional paraboloid, $\boldsymbol{\mathcal{M}}_{Pd}$, non-linearly embedded in an higher $(3(d+1))$ dimensional space, according to a multivariate Burr distribution with parameter $\alpha = 1$. Tests on this dataset are particularly challenging since the underlying manifold is characterized by a non-constant curvature.

To perform tests on datasets generated by employing a smooth non-uniform `pdf`, we propose the dataset $\boldsymbol{M}_{beta}$, obtained as follows: we sample N points in $[0,1)^{10}$, according to a beta distribution $\beta_{0.5,10}$ with parameters 0.5 and 10 respectively (high skewness), and store them in a matrix $\boldsymbol{X}_N \in \Re^{N\times 10}$; multiply each point of $\boldsymbol{X}_N$ $(\boldsymbol{X}_N(i,j))$ by $\sin(\cos(2\pi\boldsymbol{X}_N(i,j)))$, thus obtaining a matrix $\boldsymbol{D}_1 \in \Re^{N\times 10}$; multiply each point of $\boldsymbol{X}_N$ by $\cos(\sin(2\pi\boldsymbol{X}_N(i,j)))$, thus obtaining another matrix $\boldsymbol{D}_2 \in \Re^{N\times 10}$; append $\boldsymbol{D}_1$ and $\boldsymbol{D}_2$ to generate a matrix $\boldsymbol{D}_3 \in \Re^{2500\times 20}$; append $\boldsymbol{D}_3$ to its duplicate to finally generate a test dataset containing N points in $\Re^{40}$.

To further test estimators' performance on nonlinearly embedded manifolds of high `id`, we propose to generate two datasets, referred to as $\boldsymbol{M}_{N1}$ and $\boldsymbol{M}_{N2}$ in the following[7]. Precisely, to generate $\boldsymbol{M}_{N1}$ we uniformly draw N points in $[0,1]^{18}$, we transform each point by means of $\tan(\boldsymbol{x}^i \cos(\boldsymbol{x}^{18-i+1}))$ where $i = 1, \cdots, 18$, we obtain points in $\Re^{36}$ by appending each transformed $\boldsymbol{x}$ to $\arctan(\boldsymbol{x}^{18-i+1} \sin(\boldsymbol{x}^i))$, we duplicate the coordinates of each point to finally generate points in $\Re^{72}$. The `id` of $\boldsymbol{M}_{N1}$ is 18, and its points are drawn from a manifold nonlinearly embedded in $\Re^{72}$. To generate $\boldsymbol{M}_{N2}$ containing N points in $\Re^{96}$, we applied the same procedure on vectors sampled in $[0,1]^{24}$.

4.1.2 Real Datasets

Real datasets employed in literature generally concern problems in the fields of image analysis, signal processing, time series prediction, and biochemistry. Among them, the most known and used are: `ISOMAP` face database [115], `MNIST` database [71], `Isolet` dataset [38], $D2$ `Santa Fe` [93] dataset, and `DSVC1` time series [15]. Recently, the Crystal Fingerprint space for the chemical compound silicon dioxide dataset has also been proposed [119].

`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 (see Figure 2, first row).

`MNIST` database consists in 70000 gray-level images of size 28×28 of handwritten digits (see Figure 2, second row). The real `id` of this database is not actually known, but some works [49, 29] propose similar estimates for the different digits; as an example, the proposed `id` values for the digit '1' are in the range $\{8..11\}$.

`Isolet` dataset has been generated as follows: 150 subjects spoke the name of each letter of the alphabet twice, thus producing about 52 training examples from each speaker, for a total of 7797 samples. The speakers are grouped into 5 sets of 30 speakers each, and are referred to as *isolet*1, *isolet*2, *isolet*3, *isolet*4, and *isolet*5. The real `id` value characterizing this dataset is not actually known, but a study reported in [64] shows that the correct estimate could be in the range $\{16..22\}$.

The version $D2$ of `Santa Fe` dataset is a time series of 50000 one-dimensional points having nine degrees of freedom ($\texttt{id} = 9$) and being generated by a simulation of particle motion. In order to estimate the attractor dimension of this time series, it is possible to employ the method of delays described in [87], which generates D-dimensional vectors by partitioning the original dataset in blocks containing D consecutive values; as an example, by choosing $D = 50$ a dataset containing 1000 points in $\Re^{50}$ is obtained.

`DSVC1` is a time series composed by 5000 samples measured from a hardware realization of Chua's circuit [25]. Employing the method of delays with $D = 20$,

[7] A tool to generate the datasets sampled from d-dimensional paraboloids, the $\boldsymbol{M}_{beta}$ dataset, the $\boldsymbol{M}_{N1}$ dataset, and the $\boldsymbol{M}_{N2}$ dataset is available at: http://security.di.unimi.it/~fox721/dataset_generator.m

a dataset containing 250 points in $\Re^{20}$ is obtained. The `id` characterizing this dataset is ~ 2.26 [15].

Crystal Fingerprint spaces, or Crystal Fingerspaces, have been recently proposed in crystallography [119] with the aim of representing crystalline structures; these spaces are built starting from the measured distances between atoms in the crystalline structure. The theoretical `id` of one Crystal Fingerspace consists in $3N_a + 3$ crystal degrees of freedom, where N_a is the number of atoms in the crystalline unitary cell.

Figure 2: (First row) Samples from `ISOMAP` face database. (Second row) Samples from digit '0' to digit '9' in `MNIST` database.

4.2 Experimental procedures and Evaluation Measures

At the-state-of-the-art, two approaches have been mainly used to assess `id` estimators on datasets of known `id`.

The first one subsamples the test dataset to obtain T subsets of fixed cardinality and computes the percentage of correct estimations. To analyze estimators' behavior w.r.t. the cardinality of input datasets, this procedure may be repeated by using different cardinality values [49, 28, 27, 29], thus obtaining a distinct performance evaluation measure for each cardinality.

The second approach estimates the `id` on T permutations of the same dataset and averages the T `id` estimates to obtain the final one [79, 102, 73, 20]. This value is then compared with the real one to assess the `id` estimator.

To also test the estimator's robustness w.r.t. its parameter settings, in [73, 79, 102] the authors apply a further test, originally proposed by Levina et al. in [73]. Precisely, sample sets with different cardinalities are drawn from the standard Gaussian `pdf` in $\Re^5$ and, for each set, the estimator is applied varying the values of its parameters in fixed ranges; this allows to analyze the behavior of the `id` estimate as a function of both the dataset's cardinality and the parameter settings.

Note that, since `id` estimators are usually tested on different datasets to evaluate their reliability when confronted by different dataset structures and configurations, in [79] an overall evaluation measure is proposed. This indicator, called Mean Percentage Error (`MPE`), summarizes all the obtained results in a unique value computed as: $\texttt{MPE} = \frac{100}{\#\boldsymbol{M}} \sum_{\boldsymbol{M}} \frac{|\hat{d}_M - d_M|}{d_M}$, where $\#\boldsymbol{M}$ is the number of tested datasets, $\hat{d}_{\boldsymbol{M}}$ is the id estimated on the dataset $\boldsymbol{M}$, and $d_{\boldsymbol{M}}$ is the real id of $\boldsymbol{M}$. To apply this technique to real datasets whose id belongs to the range $\{d_{min}..d_{max}\}$, the same authors propose to calculate the associated MPE's term as: $\min_{d\in\{d_{min}..d_{max}\}}\left(\frac{|\hat{d}_{\boldsymbol{M}}-d|}{d_{\boldsymbol{M}}}\right)$, where $d_{\boldsymbol{M}}$ is the mean of the range.

4.3 Benchmark

In this section we propose an evaluation approach which can be used as a standard framework to assess estimators performance, comparing it to relevant id estimators whose code is publicly available. In this benchmark, we suggest to use the following estimators (see Section 3): Hein, MLE, kNNG, MLSVD, BPCA, CD, MiND_{KL}, and DANCo[8]. Note that these estimators cover all the groups described in Section 3, that is *Projective*, *Fractal*, *Nearest-Neighbors based*, *Graph based* estimators.

The benchmark is composed by following steps:

1. Test all the considered estimators on both the synthetic and real datasets described below. We highlight that the synthetic datasets whose id is a user-defined parameter should be created with sufficiently high id values (id ≥ 10).

2. Comparative Evaluation steps:

   a) compute the MPE indicator both for synthetic and real datasets.

   b) compute a ranking test with control methods; to this aim, we suggest the Friedman test with Bonferroni-Dunn post-hoc analyses [54].

   c) perform the tests proposed in [73] to evaluate the robustness, w.r.t different cardinalities and parameter settings.

The 21 synthetic datasets used in the benchmark, referred to as $\boldsymbol{M}_*$ in the following, are listed in Table 2 with their relevant characteristics (N, d, and D). The first 15 datasets are generated with the tool proposed in [49]; they include 4 instances, $\boldsymbol{M}_{10*}$, of dataset $\boldsymbol{M}_{10}$, which are drawn from $\boldsymbol{\mathcal{M}}_{10}^{H}$ after its embedding in $\Re^D$ by setting $D = \{11, 18, 25, 71\}$. Note that we did not include the dataset sampled from $\boldsymbol{\mathcal{M}}_{8}^{H}$ (see Table 1) since relevant and recent id estimators have similarly produced highly overestimated results when tested on it [102]. Indeed, dealing with highly curved manifolds is still a quite challenging problem in the field.

[8] The source code of the mentioned methods is available at:
Hein: http://www.ml.uni-saarland.de/code.shtml,
MLE: http://www.stat.lsa.umich.edu/~elevina/mledim.m,
kNNG: http://www.eecs.umich.edu/~hero/IntrinsicDim/,
MLSVD: http://www.math.duke.edu/~mauro/code.html♯MSVD,
BPCA: http://research.microsoft.com/en-us/um/cambridge/projects/infernet/blogs/bayesianpca.aspx
CD: http://cseweb.ucsd.edu/~lvdmaaten/dr/download.php,
MiND_{KL}, and DANCo: http://www.mathworks.it/matlabcentral/fileexchange/40112-intrinsic-dimensionality-estimation-techniques

The last six synthetic datasets are $\boldsymbol{M}_{N1}$, $\boldsymbol{M}_{N2}$, $\boldsymbol{M}_{beta}$, and 3 instances of dataset $\boldsymbol{M}_{P*}$, which are sampled from paraboloids $\boldsymbol{\mathcal{M}}_{Pd}$ whose `id` is, respectively, $d = \{3, 6, 9\}$.

To perform multiple tests, 20 instances of each dataset have been generated, and the achieved results have been averaged.

Dataset	**Dataset Name**	$\boldsymbol{N}$	$\boldsymbol{d}$	$\boldsymbol{D}$
Syntethic	$\boldsymbol{M}_1$	2500	10	11
	$\boldsymbol{M}_2$	2500	3	5
	$\boldsymbol{M}_3$	2500	4	6
	$\boldsymbol{M}_4$	2500	4	8
	$\boldsymbol{M}_5$	2500	2	3
	$\boldsymbol{M}_6$	2500	6	36
	$\boldsymbol{M}_7$	2500	2	3
	$\boldsymbol{M}_9$	2500	20	20
	$\boldsymbol{M}_{10a}$	2500	10	11
	$\boldsymbol{M}_{10b}$	2500	17	18
	$\boldsymbol{M}_{10c}$	2500	24	25
	$\boldsymbol{M}_{10d}$	2500	70	71
	$\boldsymbol{M}_{11}$	2500	2	3
	$\boldsymbol{M}_{12}$	2500	20	20
	$\boldsymbol{M}_{13}$	2500	1	13
	$\boldsymbol{M}_{N1}$	2500	18	72
	$\boldsymbol{M}_{N2}$	2500	24	96
	$\boldsymbol{M}_{beta}$	2500	10	40
	$\boldsymbol{M}_{P3}$	2500	3	12
	$\boldsymbol{M}_{P6}$	2500	6	21
	$\boldsymbol{M}_{P9}$	2500	9	30
Real	$\boldsymbol{M}_{\texttt{DSCV1}}$	250	2.26	20
	$\boldsymbol{M}_{\texttt{ISOMAP}}$	698	3.00	4096
	$\boldsymbol{M}_{\texttt{SantaFe}}$	1000	9.00	50
	$\boldsymbol{M}_{\texttt{MNIST1}}$	70000	$8.00 - 11.00$	784
	$\boldsymbol{M}_{\texttt{SiO2}}$	4738	12.00	1800
	$\boldsymbol{M}_{\texttt{Isolet}}$	7797	$16.00 - 22.00$	617

Table 2: Synthetic datasets and real datasets suggested by the benchmark; $\boldsymbol{N}$ is the dataset cardinality, $\boldsymbol{d}$ is the `id`, and $\boldsymbol{D}$ is the embedding space dimension.

Regarding the real datasets we used the `DSVC1` time series [15] ($\boldsymbol{M}_{\texttt{DSVC1}}$, `id` ~ 2.26), the `ISOMAP` face database [115] ($\boldsymbol{M}_{\texttt{ISOMAP}}$, `id` $= 3$), the `Santa Fe` dataset [93] ($\boldsymbol{M}_{\texttt{SantaFe}}$, `id` $= 9$), the `MNIST` database [71] ($\boldsymbol{M}_{\texttt{MNIST1}}$, `id` $\in \{8..13\}$), the `Isolet` dataset [38] ($\boldsymbol{M}_{\texttt{Isolet}}$, `id` $\in \{16..22\}$), and the Crystal Fingerprint space for the chemical compound silicon dioxide SiO_2 structure with 3 atoms (this allows to obtain the $\boldsymbol{M}_{\texttt{SiO2}}$ dataset containing 4738 points embedded in $\Re^{1800}$, and being characterized by an `id` equal to 12).

To run multiple tests also on $\boldsymbol{M}_{\texttt{MNIST1}}$, $\boldsymbol{M}_{\mathrm{SiO}_2}$, and $\boldsymbol{M}_{\texttt{Isolet}}$, for each of them we generated 5 instances by extracting random subsets containing 2500 points each and we averaged the achieved results.

Table 3 summarizes the parameter values we employed for different estimators. Note that, to relax the dependency of the `kNNG` algorithm from the setting

Dataset	Method	Parameters
Synthetic	MLE	$k_1 = 6\ k_2 = 20$
	DANCo	$k = 10$
	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	$iters = 2000,\ \alpha = (2.0, 2.0)\ \pi = (2.0, 2.0)\ \mu = (0.0, 0.01)$
	Hein	$None$
	CD	$None$
	MLSVD	$None$
	MiND$_{\text{KL}}$	$k = 10$
Real	MLE	$k_1 = 3\ k_2 = 8$
	DANCo	$k = 5$
	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	$iters = 2000,\ \alpha = (2.0, 2.0)\ \pi = (2.0, 2.0)\ \mu = (0.0, 0.01)$
	Hein	$None$
	CD	$None$
	MLSVD	$None$
	MiND$_{\text{KL}}$	$k = 5$

Table 3: 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 re-sampling trials per LS iteration, α and π represent the parameters (shape and rate) of the Gamma prior distributions, which describe the hyper-parameters 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.

of its parameter k, we performed multiple runs with $k_1 \leq k \leq k_2$ and we averaged the achieved results. Furthermore, we tested two versions of the algorithm (referred to as kNNG$_1$ and kNNG$_2$) obtained by varying the parameters M and N.

4.4 Experimental Results

Table 4 summarizes the results obtained by the compared estimators on the synthetic datasets, while in Table 5 the results obtained on the real datasets are reported.

Looking at the number of correct estimations computed by each algorithm (highlighted in boldface), we have the following ranking: MLSVD is correct on 13 out of 21 synthetic datasets, DANCo (correct on 10 out of 21 datasets), Hein (correct on 6 out of 21), MiND$_{\text{KL}}$ (6 out of 21), BPCA (4 out of 21), and MLE (1 out of 21). It can be further noted that kNNG$_*$, CD, MLE, and Hein obtain good estimates only for low id manifolds, while they produce underestimated values when processing datasets of high id.

By observing the MPE indicator, which accounts for the precision of the achieved estimates, we obtain a different ranking: DANCo, MiND$_{\text{KL}}$, kNNG$_1$ and kNNG$_2$, MLE, Hein, CD, and MLSVD. This difference is due to the fact that algorithms, such as kNNG$_1$ and kNNG$_2$, MLE, and Hein, most of the times produce

Dataset	d	MLE	kNNG$_1$	kNNG$_2$	BPCA	Hein	CD	MiND$_{\text{KL}}$	DANCo	MLSVD
$\boldsymbol{M}_1$	*10.00*	9.10	9.16	9.89	5.45	9.45	9.12	10.30	10.09	**10.00**
$\boldsymbol{M}_2$	*3.00*	2.88	2.95	3.03	**3.00**	**3.00**	2.88	**3.00**	**3.00**	**3.00**
$\boldsymbol{M}_3$	*4.00*	3.83	3.75	3.82	**4.00**	**4.00**	3.23	**4.00**	**4.00**	2.08
$\boldsymbol{M}_4$	*4.00*	3.95	4.05	4.76	4.25	**4.00**	3.88	4.15	**4.00**	8.00
$\boldsymbol{M}_5$	*2.00*	1.97	1.96	2.06	**2.00**	**2.00**	1.98	**2.00**	**2.00**	**2.00**
$\boldsymbol{M}_6$	*6.00*	6.39	6.46	11.24	12.00	**5.95**	5.91	6.50	7.00	12.00
$\boldsymbol{M}_7$	*2.00*	1.96	1.97	2.09	**2.00**	**2.00**	1.93	2.07	**2.00**	2.35
$\boldsymbol{M}_9$	*20.00*	14.64	15.25	10.59	13.55	15.50	13.75	19.15	19.71	**20.00**
$\boldsymbol{M}_{10a}$	*10.00*	8.26	8.62	10.21	5.20	8.90	8.09	9.85	9.86	**10.00**
$\boldsymbol{M}_{10b}$	*17.00*	12.87	13.69	15.38	9.46	13.85	12.30	16.25	16.62	**17.00**
$\boldsymbol{M}_{10c}$	*24.00*	16.96	17.67	21.42	13.3	17.95	15.58	22.55	24.28	**24.00**
$\boldsymbol{M}_{10d}$	*70.00*	36.49	39.67	40.31	71.00	38.69	31.4	64.38	70.52	**70.00**
$\boldsymbol{M}_{11}$	*2.00*	2.21	1.95	2.03	1.55	2.00	2.19	**2.00**	**2.00**	1.00
$\boldsymbol{M}_{12}$	*20.00*	15.82	16.40	24.89	13.7	15.00	11.26	19.35	19.90	**20.00**
$\boldsymbol{M}_{13}$	*1.00*	**1.00**	0.97	1.07	5.70	**1.00**	1.14	**1.00**	**1.00**	**1.00**
$\boldsymbol{M}_{N1}$	*18.00*	12.25	14.26	19.8	36.00	14.10	10.40	17.76	18.76	**18.00**
$\boldsymbol{M}_{N2}$	*24.00*	14.72	17.62	26.87	48.00	17.76	12.43	23.76	25.76	**24.00**
$\boldsymbol{M}_{beta}$	*10.00*	6.36	6.45	14.77	19.7	4.00	3.05	7.00	7.00	**10.00**
$\boldsymbol{M}_{P3}$	*3.00*	2.89	2.93	3.12	7.00	2.00	2.43	**3.00**	**3.00**	1.00
$\boldsymbol{M}_{P6}$	*6.00*	4.96	4.98	5.82	7.00	2.66	3.58	5.04	**6.00**	1.00
$\boldsymbol{M}_{P9}$	*9.00*	6.35	6.89	8.04	10.95	2.85	4.55	7.00	**8.00**	1.00
	MPE	17.29	14.50	16.79	62.62	19.92	25.96	5.55	**3.70**	26.34

Table 4: Results achieved on the synthetic datasets. The bottom row reports the MPE achieved by each algorithm; anyhow, for each test dataset the best approximation results are highlighted in boldface.

Dataset	id	MLE	kNNG$_1$	kNNG$_2$	BPCA	Hein	CD	MiND$_{\text{KL}}$	DANCo	MLSVD
$\boldsymbol{M}_{\texttt{DSCV1}}$	*2.26*	2.03	1.77	1.86	6.00	3.00	1.92	2.50	**2.26**	1.75
$\boldsymbol{M}_{\texttt{ISOMAP}}$	*3.00*	4.05	3.60	4.32	4.00	**3.00**	3.37	3.9	4.00	1.00
$\boldsymbol{M}_{\texttt{SantaFe}}$	*9.00*	7.16	7.28	7.43	18.00	6.00	4.39	7.60	**8.19**	1.00
$\boldsymbol{M}_{\texttt{MNIST1}}$	*8.00-11.00*	10.29	10.37	9.58	11.00	8.00	6.96	11.00	9.98	1.00
$\boldsymbol{M}_{\texttt{SiO2}}$	*12.00*	39.28	10.24	10.36	3.00	4.80	1.05	17.20	**12.60**	1.00
$\boldsymbol{M}_{\texttt{Isolet}}$	*16.00-22.00*	15.78	6.50	8.32	19.00	3.00	3.65	20.00	19.00	1.00
	MPE	53.83	27.41	26.76	71.68	34.50	43.34	27.00	**15.14**	75.17

Table 5: Results achieved on the real datasets by the compared approaches. The bottom row reports the MPE achieved by each algorithm; anyhow, for each test dataset the best approximation results are highlighted in boldface (when the real id takes values in a range, we highlighted the results that best approximate the mean value of the range).

results close to the correct value.

Regarding the real datasets, all the algorithms achieve a much worse MPE indicator, and again DANCo is the best performing.

Furthermore, we compute the Friedman ranking test with the Bonferroni-Dunn post-hoc analysis as proposed in Section 4.3 to state the quality of the achieved results on both the synthetic and real datasets. Table 6 and Table 7 summarize the ranking results.

Finally, we performed the tests proposed in [73] to evaluate the robustness of MiND$_{\text{KL}}$, MLE, DANCo, and kNNG$_*$ w.r.t. the settings of their k parameter. Precisely, these tests employ synthetic datasets sub-sampled from the standard Gaussian pdf in $\Re^5$ (id $= 5$). As proposed in Section Section 4.2, we repeated the tests for datasets with cardinalities $N \in \{200, 500, 1000, 2000\}$ varying the

Method	Ranking
DANCo	2.40
MiND$_{KL}$	3.46
Hein	4.67
kNNG$_2$	5.11
MLSVD	5.17
kNNG$_1$	5.17
MLE	5.70
CD	6.63
BPCA	6.68

Table 6: Friedman Ranking results achieved on all the datasets. The null hypothesis that the algorithms perform comparably is rejected with p-value < 0.00001.

	MiND$_{KL}$	Hein	kNNG$_1$	kNNG$_2$	MLE	CD	MLSVD	BPCA
DANCo	0.1567	0.0024	0.0003	0.0002	0.0002	0.0000	0.0000	0.0000
MiND$_{KL}$	***	0.0801	0.0303	0.0244	0.0055	0.0020	0.0000	0.0000
Hein	***	***	0.7528	0.6366	0.1564	0.1474	0.0034	0.0018
kNNG$_1$	***	***	***	0.8557	0.3443	0.2301	0.0164	0.0071
kNNG$_2$	***	***	***	***	0.9314	0.3894	0.1113	0.0282
MLE	***	***	***	***	***	0.3428	0.1876	0.0307
CD	***	***	***	***	***	***	0.7337	0.1961

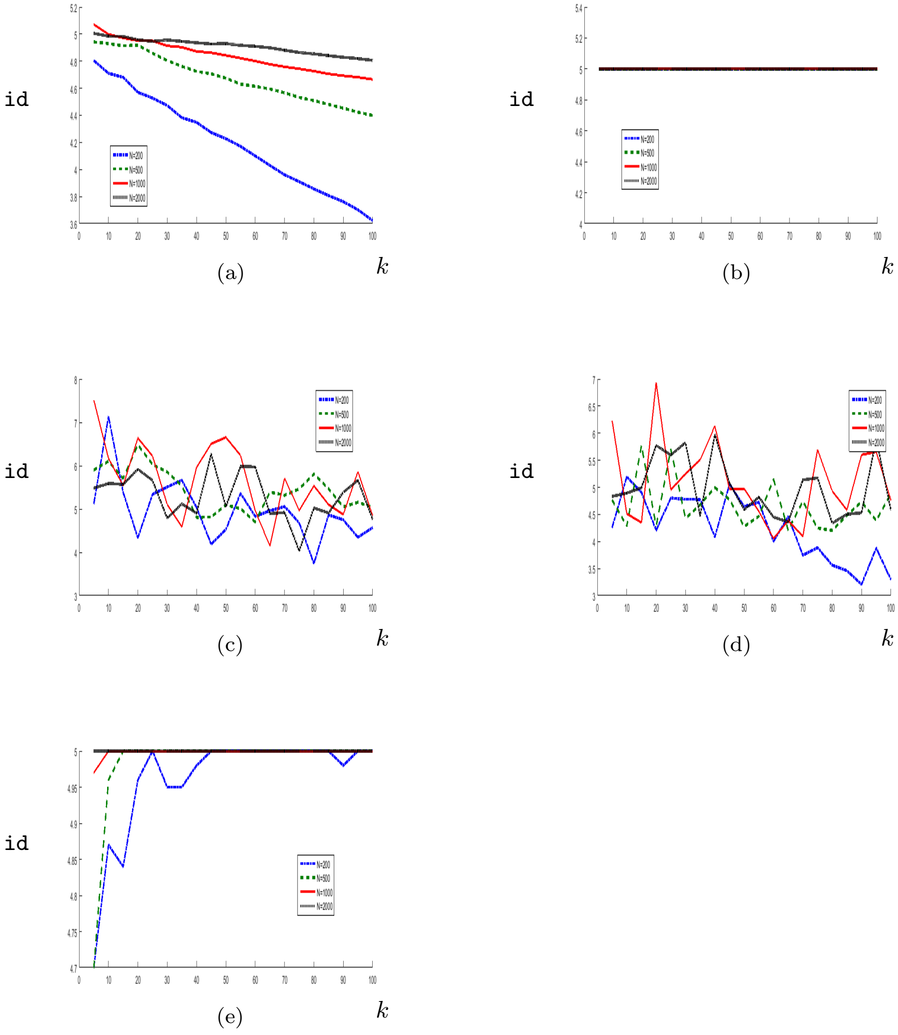
Table 7: Hypothesis testing of significance between techniques. Bonferroni-Dunn‘s procedure rejects those hypotheses that have a p-value≤ 0.0125.

parameter k in the range $\{5..100\}$.
As shown in Figure 3 many of the tested techniques are strongly influenced by the parameter settings; therefore, studying the variability of the algorithms' behavior when changing their parameter settings is of utmost importance.

5 Conclusions and Open Problems

This work presents the base theories of state of the art id estimators and surveys the most relevant and recent among them, highlighting their strengths and their drawbacks.

Unfortunately, performing an objective comparative evaluation among the surveyed methods is difficult because, to our knowledge, no benchmark framework exists in this research field; therefore, in Section 4 we propose an evaluation approach that employs both real and synthetic datasets, and suggests experiments to evaluate the estimators' robustness w.r.t. their parameter settings. Note that, the benchmark is designed to evaluate the performance achieved by id estimators when both low and high id data must be processed; this consideration is due to the fact that, to our knowledge, only few methods [16, 79, 102, 20] have empirically investigated the problem of datasets drawn from manifolds non-linearly embedded in higher dimensional spaces and characterized by a sufficiently high id (that is id $\geqslant 10$). However, due to the continuous technological advances, high id datasets are becoming more and more common, ad the construction of a theoretically well-formed and robust id estimator able to deal with high id data and limited amount of points remains one of the open research challenges in machine learning. Besides, id estimators should be developed by also



(a)

(b)

(c)

(d)

(e)

Figure 3: Behavior of: (a) MLE, (b) DANCo , (c) $\mathtt{kNNG}_1$, (d) $\mathtt{kNNG}_2$, and (e) $\mathtt{MiND}_{\mathtt{KL}}$ applied to points drawn from a 5-dimensional standard normal distribution; in this test $N \in \{200, 500, 1000, 2000\}$ and $k \in \{5..100\}$.

considering datasets drawn through non-uniform smooth pdfs from manifolds $\mathcal{M}$ characterized by a non-constant curvature; indeed, most of the algorithms are tested by only employing data drawn by means of uniform pdf.

We further note that, though the aforementioned problems still need further investigations, most researches in this field are presently focusing on tasks that require to estimate the id as the first step. Examples are: “multi-manifold learning”, whose aim is to process datasets drawn from multiple manifolds, each characterized by different id, to identify the underlying structures (see [44] for an example); “non-linear dimensionality reduction” or “manifold reconstruction”, whose aim is to find the mapping that projects the data (embedded in a higher D-dimensional space) on the lower $\hat{d}$-dimensional subspace, being $\hat{d}$ the id estimated on the input dataset (as examples, see [124, 43, 127]).

References

[1] Y. Ashkenazy. The use of generalized information dimension in measuring fractal dimension of time series. *Physica A: Statistical Mechanics and its Applications*, 271(34):427 – 447, 1999.

[2] R. E. Bellman. *Adaptive Control Processes: A Guided Tour.* Princeton University Press, 1961.

[3] R. S. Bennett. The Intrinsic Dimensionality of Signal Collections. *IEEE Trans. on Information Theory*, IT-15(5):517–525, 1969.

[4] K.S. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft. When is ”nearest neighbor” meaningful? In *Proceedings of the 7th International Conference on Database Theory*, ICDT ’99, pages 217–235, London, UK, UK, 1999. Springer-Verlag.

[5] P.J. Bickel and D. Yan. Sparsity and the possibility of inference. *Sankhya The Indian Journal of Statistics*, 70(1):0–23, 2008.

[6] C. M. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, Oxford, 1995.

[7] C. M. Bishop. Bayesian PCA. *Proc. of NIPS*, 11:382–388, 1998.

[8] C. M. Bishop. *Pattern recognition and machine learning.* Springer, 1st ed. 2006. corr. 2nd printing edition, October 2006.

[9] C. Bouveyron, G. Celeux, and S. Girard. Intrinsic dimension estimation by maximum likelihood in isotropic probabilistic pca. *Pattern Recognition Letters*, 32(14):1706–1713, 2011.

[10] M.C. Brito, A.J. Quiroz, and J.E. Yukich. Graph theoretic procedures for dimension identification. *Journal of Multivariate Analysis*, 81:67–84, 2002.

[11] M.C. Brito, A.J. Quiroz, and J.E. Yukich. Intrinsic dimension identification via graph-theoretic methods. *Journal of Multivariate Analysis*, 116:263–277, 2013.

[12] L. Brouwer. *Collected Works*, volume I, Philosophy and foundations of mathematics and II, Geometry, Analysis, Topology and Mechanics. North Holland/American Elsevier, 1976.

[13] J. Bruske and G. Sommer. Intrinsic dimensionality estimation with optimally topology preserving maps. *IEEE Trans. on PAMI*, 20(5):572–575, 1998.

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

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

[16] F. Camastra and A. Vinciarelli. Estimating the intrinsic dimension of data with a fractal-based method. *IEEE Trans. on PAMI*, 24:1404–1407, 2002.

[17] P. Campadelli, E. Casiraghi, C. Ceruti, G. Lombardi, and A. Rozza. Local intrinsic dimensionality based features for clustering. In Alfredo Petrosino, editor, *ICIAP (1)*, volume 8156 of *Lecture Notes in Computer Science*, pages 41–50. Springer, 2013.

[18] K. Carter, R. Raich, and A. O. Hero. On local intrinsic dimension estimation and its applications. *IEEE Trans. on Signal Processing*, 58(2):650–663, 2010.

[19] K.M. Carter, R. Raich, W.G. Finn, and A.O.III Hero. Fine: Fisher information nonparametric embedding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(11):2093–2098, 2009.

[20] C. Ceruti, S. Bassis, A Rozza, G. Lombardi, E. Casiraghi, and P. Campadelli. DANCo: an intrinsic Dimensionalty estimator exploiting Angle and Norm Concentration. *Pattern recognition*, 2014.

[21] Edgar Chávez, Gonzalo Navarro, Ricardo Baeza-Yates, and José Luis Marroquín. Searching in metric spaces. *ACM Comput. Surv.*, 33(3):273–321, 2001.

[22] C. K. Chen and H. C. Andrews. Nonlinear intrinsic dimensionality computations. *IEEE Trans. on System Man and Cybernetics*, SMC-3:197–200, 1973.

[23] M. Chen, J. Silva, J. Paisley, C. Wang, D. Dunson, and L. Carin. Compressive sensing on manifolds using a nonparametric mixture of factor analyzers: Algorithm and performance bounds. *IEEE Transactions on Signal Processing*, 58(12):6140–6155, 2010.

[24] D. Chialvo, R. Gilmour, and J. Jalife. Low dimensional chaos in cardiac tissue. *Nature*, 343:653–657, 1990.

[25] L. Chua, M. Komuro, and T. Matsumoto. The double scroll. *IEEE Trans. on Circuits and Systems*, 32:797–818, 1985.

[26] A. Costa, J. A. Girotra and A. O. Hero. Estimating local intrinsic dimension with k-nearest neighbor graphs. *in: IEEE/SP 13th Workshop on Statistical Signal Processing, IEEE Conference Publication*, pages 417–422, 2005.

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

[28] J. A. Costa and A. O. Hero. Learning intrinsic dimension and entropy of high-dimensional shape spaces. In *Proc. of EUSIPCO*, pages 231–252, 2004.

[29] J. A. Costa and A. O. Hero. *Determining intrinsic dimension and entropy of high-dimensional shape spaces.* Boston, MA: Birkhäuser, 2006.

[30] M. Das Gupta and T. S. Huang. Regularized maximum likelihood for intrinsic dimension estimation. In P. Grünwald and P. Spirtes, editors, *UAI*, pages 220–227. AUAI Press, 2010.

[31] P. Demartines and J. Herault. Curvilinear component analysis: A self-organizing neural network for nonlinear mapping in cluster analysis. *IEEE Trans. on Neural Networks*, 8(1):148–154, 1997.

[32] N.G. Derry and S.P. Derry. Age dependence of the menstrual cycle correlation dimension. *Open Journal of Biophysics*, 2:40–45, 2012.

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

[34] R. Everson and S. Roberts. Inferring the eigenvalues of covariance matrices from limited, noisy data. *IEEE Trans. Sig. Proc*, 2000.

[35] K. Falconer. *Fractal Geometry - Mathematical Foundations and Applications.* John Wiley, Second Edition, 2003.

[36] M. Fan, H. Qiao, and B. Zhang. Intrinsic dimension estimation of manifolds by incising balls. *Pattern Recogn.*, 42(5):780–787, May 2009.

[37] A. M. Farahmand, C. Szepesvari, and J. Y. Audibert. Manifold-adaptive dimension estimation. *Proc. of ICML*, pages 265–272, 2007.

[38] A. Frank and A. Asuncion. UCI machine learning repository, 2010.

[39] J. H. Friedman and L.C. Rafsky. Graph theoretic measures of multivariate association and prediction. *Annals of Statistics*, 11:377–391, 1983.

[40] J.H. Friedman, T. Hastie, and R. Tibshirani. *The Elements of Statistical Learning - Data Mining, Inference and Prediction.* Springer, Berlin, 2009.

[41] K. Fukunaga and D.R. Olsen. An algorithm for finding intrinsic dimensionality of data. *IEEE Trans. on Computers*, C-20(2):176–183, 1971.

[42] M. Gashler and T. Martinez. Robust manifold learning with cyclecut. *Connection Science*, 2011.

[43] M. Gashler and T. Martinez. Tangent space guided intelligent neighbor finding. *Proceedings of International Joint Conference on Neural Networks*, 2011.

[44] D. Gong, X. Zhao, and G. Medioni. Robust Multiple Manifolds Structure Learning. *Proceedings of the International Conference on Machine Learning, Edinburgh, Scotland*, 2012.

[45] P. Grassberger and I. Procaccia. Measuring the strangeness of strange attractors. *Physica D: Nonlinear Phenomena*, 9:189–208, 1983.

[46] Y. Guan and J. G. Dy. Sparse probabilistic principal component analysis. *J. of Machine Learning Research - Proc. Track*, 5:185–192, 2009.

[47] G. Haro, G. Randall, and G. Sapiro. Translated poisson mixture model for stratification learning. *International Journal on Computer Vision*, 80(3):358 –374, 2008.

[48] S. Haykin and Xiao Bo Li. Detection of signals in chaos. *Proceedings of the IEEE*, 83(1):95 –122, jan 1995.

[49] M. Hein and J.Y. Audibert. Intrinsic dimensionality estimation of submanifolds in euclidean space. In *Proc. of ICML*, pages 289–296, 2005.

[50] O. A. Hero, B. Ma, O. Michel, and J. Gorman. Applications of entropic spanning graphs. *IEEE Signal Processing Magazine*, 19(5):85–95, 2002.

[51] R. Heylen and P. Scheunders. Hyperspectral Intrinsic Dimensionality Estimation with Nearest-Neighbor Distance Ratios. *IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING*, 6(2):570–579, 2013.

[52] B. Hu, T. Rakthanmanon, Y. Hao, S. Evans, S. Lonardi, and E. Keogh. Towards Discovering the Intrinsic Cardinality and Dimensionality of Time Series Using MDL. *Algorithmic Probability and Friends. Bayesian Prediction and Artificial Intelligence*, LNCS 7070:184 – 197, 2013.

[53] Valerie Isham. *Statistical aspects of chaos: A review.* London: Chapman and Hall, 1993.

[54] J Jaccard, M. A. Becker, and G. Wood. Pairwise multiple comparison procedures: A review. *Psychological Bulletin*, 96(3):589–596, 1984.

[55] I.M. James. *History of Topology.* Mathematics. Elsevier, 1999.

[56] I. T. Jollife. *Principal Component Analysis.* Springer Series in Statistics. Springer-Verlag, New York, NY, 1986.

[57] Kumaraswamy K., Vasileios Megalooikonomou, and Christos Faloutsos. Fractal dimension and vector quantization. *Inf. Process. Lett.*, 91(3):107–113, 2004.

[58] R. Karbauskaite and G. Dzemyda. Investigation of the maximum likelihood estimator of intrinsic dimensionality. *Proceedings of the 10th International Conference on Computer Data Analysis and Modeling, Minsk*, 2:110–113, 2013.

[59] R. Karbauskaite, G. Dzemyda, and E. Mazetis. Geodesic distances in the maximum likelihood estimator of intrinsic dimensionality. *Non linear Analysis: Modelling and Control*, 16:387402, 2011.

[60] F.G. Kaslovsky, D.N.and Meyer. OPTIMAL TANGENT PLANE RECOVERY FROM NOISY MANIFOLD SAMPLES. *CoRR–arXiv*, 2011.

[61] M. Katetov and P. Simon. Origins of dimension theory. *Handbook of the History of General Topology*, 1997.

[62] B. Kégl. Intrinsic dimension estimation using packing numbers. In S. Becker, S. Thrun, and K. Obermayer, editors, *Proc. of NIPS*, pages 681–688. MIT Press, 2002.

[63] M. Kirby. *Geometric Data Analysis: an Empirical Approach to Dimensionality Reduction and the Study of Patterns.* John Wiley and Sons, 1998.

[64] I. Kivimäki, K. Lagus, I. Nieminen, J. Väyrynen, and T. Honkela. Using correlation dimension for analysing text data. In *Proc. of the ICANN*, pages 368–373. Springer-Verlag, 2010.

[65] J.B. Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29:1–27, 1964.

[66] J.B. Kruskal. *Linear transformation of multivariate data to reveal clustering*, volume I. Ac. Press, 1972.

[67] J.B. Kruskal and J. D. Carrol. *Geometrical models and badness-of-fit functions*, volume 2. Ac. Press, 1969.

[68] H. Lähdesmäki, O. Yli-Harja, W. Zhang, and I. Shmulevich. Intrinsic dimensionality in gene expression analysis. In *Proceedings of GENSIPS*, pages –, 2005.

[69] J. Lapuyade-Lahorgue and A. Mohammad-Djafari. Nearest neighbors and correlation dimension for dimensionality estimation. Application to factor analysis of real biological time series data. *Proceedings of European Symposium on Artificial Neural Networks (ESANN11)*, 2011.

[70] D.C Laughlin. The intrinsic dimensionality of plant traits and its relevance to community assembly. *Journal of Ecology*, 102:186–193, 2014.

[71] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proc. of IEEE*, 86:2278–2324, 1998.

[72] J. A. Lee and M. Verleysen. *Nonlinear dimensionality reduction.* Springer, 2007.

[73] E. Levina and P. J. Bickel. Maximum likelihood estimation of intrinsic dimension. *Proceedings of NIPS*, 1:777–784, 2004.

[74] C. Li, J. Guo, and B. Xiao. Intrinsic dimensionality estimation within neighborhood convex hull. *International Journal of Pattern Recognition and Artificial Intelligence*, 23(01):31–44, 2009.

[75] J. Li and D. Tao. Simple exponential family PCA. *Proc. of AISTATS*, pages 453–460, 2010.

[76] T. Lin and H. Zha. Riemannian manifold learning. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 30(5):796–809, 2008.

[77] A. V. Little, M. Maggioni, and L. Rosasco. Multiscale Geometric Methods for Data Sets I: Multiscale SVD, Noise and Curvature. *MIT-CSAIL-TR-2012-029*, 2012.

[78] G. Lombardi, E. Casiraghi, and P. Campadelli. Curvature Estimation and Curve Inference with Tensor Voting: a New Approach. *Proc. of ACIVS 2008*, 5259:613–624, 2008.

[79] G. Lombardi, A. Rozza, C. Ceruti, E. Casiraghi, and P. Campadelli. Minimum neighbor distance estimators of intrinsic dimension. *Proc. of ECML-PKDD*, 6912:374–389, 2011.

[80] D. MacKay and Z. Ghahramani. Comments on maximum likelihood estimation of intrinsic dimension by E. Levina and P. Bickel, 2005. http://www.inference.phy.cam.ac.uk/mackay/dimension/.

[81] K. V. Mardia. *Statistics of Directional Data.* Ac. Press, 1972.

[82] T. Martinetz and K. Schulten. Topology representing networks. *Neural Networks*, 3:507–522, 1994.

[83] G. Medioni and P. Mordohai. The tensor voting framework. *Emerging Topics in Computer Vision*, pages 191–255, 2004.

[84] A. Mekler. Calculation of eeg correlation dimension: Large massifs of experimental data. *Computer Methods and Programs in Biomedicine*, 92(1):154 – 160, 2008.

[85] T. P. Minka. Automatic choice of dimensionality for PCA. Technical Report 514, MIT, 2000.

[86] P. Mordohai and G. Medioni. Dimensionality estimation, manifold learning and function approximation using tensor voting. *Journal of Machine Learning Research*, 11:411–450, March 2010.

[87] E. Ott. *Chaos in Dynamical Systems.* Cambridge University Press, Cambridge, 1993.

[88] M.D. Penrose and J.E. Yukich. Central limit theorems for some graphs in computational geometry. *The Annals of Applied Probability*, 11:1005–1041, 2001.

[89] M.D. Penrose and J.E. Yukich. Limit theory for point processes in manifolds. *The Annals of Applied Probability (in press)*, arXiv:1104.0914, 2012.

[90] V. Pestov. An axiomatic approach to intrinsic dimension of a dataset. *Neural Networks*, 21(23):204 – 213, 2008.

[91] V. Pestov. Intrinsic dimensionality. *SIGSPATIAL Special*, 2(2):8–11, 2010.

[92] K. Pettis, T. Bailey, A. Jain, and R. Dubes. An intrinsic dimensionality estimator from near-neighbor information. *IEEE Trans. on PAMI*, 1(1):25–37, 1979.

[93] F. Pineda and J. Sommerer. Estimating generalized dimensions and choosing time delays: A fast algorithm. *Time Series Prediction. Forecasting the Future and Understanding the Past*, pages 367–385, 1994.

[94] M Polito and P. Perona. Grouping and dimensionality reduction by locally linear embedding. *Advances in Neural Information Processing Systems*, 14:1255–1262, 2001.

[95] A.J. Quiroz. *Graph-theoretical methods. in: Encyclopedia of Statistical Sciences*, volume 5. Wiley and Sons, New York, 2006.

[96] M. Raginsky and S. Lazebnik. Estimation of intrinsic dimensionality using high-rate vector quantization. In *NIPS*, pages 1105–1112, 2005.

[97] J.J. Rajan and P.J.W. Rayner. Model order selection for the singular-value decomposition and the discrete karhunen-loeve transform using a bayesian-approach. *VISP*, 144(2):116–123, April 1997.

[98] J.C. Robinson. *Dimensions, Embeddings, and Attractors*. CAMBRIDGE TRACTS IN MATHEMATICS. Cambridge University Press, 2010.

[99] A. K. Romney, R. N. Shepard, and S. B. Nerlove. *Multidimensionaling Scaling*, volume I, Theory. Seminar Press, 1972.

[100] A. K. Romney, R. N. Shepard, and S. B. Nerlove. *Multidimensionaling Scaling*, volume II, Applications. Seminar Press, 1972.

[101] S.T. Roweis and L.K. Saul. Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science*, 290:2323–2326, December 2000.

[102] A. Rozza, G. Lombardi, C. Ceruti, E. Casiraghi, and P. Campadelli. Novel high intrinsic dimensionality estimators. *Machine Learning Journal*, 89(1-2):37–65, May 2012.

[103] J. W. J. Sammon. A nonlinear mapping for data structure analysis. *IEEE Trans. on Computers*, C-18:401–409, 1969.

[104] J. A. Scheinkman and B. Lebaron. Nonlinear dynamics and stock returns. *The Journal of Business*, 62(3):311–37, 1989.

[105] B. Schölkopf, A. Smola, and K. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Comput.*, 10(5):1299–1319, July 1998.

[106] R.N. Shepard. The analysis of proximities: Multidimensional scaling with an unknown distance function, part I. *Psychometrika*, 27:125–140, 1962.

[107] R.N. Shepard. The analysis of proximities: Multidimensional scaling with an unknown distance function, part II. *Psychometrika*, 27:19–246, 1962.

[108] R.N. Shepard and J.D. Carroll. *Parametric representation of nonlinear data structures*. Ac. Press, 1969.

[109] M.F. Shilling. Mutual and shared neighbor probabilities: finite and infinite dimensional results. *Advances in Applied Probability*, 18:388–405, 1986.

[110] P. Somervuo. Speech dimensionality analysis on hypercubical self-organizing maps. *Neural Process. Lett.*, 17(2):125–136, April 2003.

[111] K. Sricharan, R. Raich, and A.O. Hero. Optimized intrinsic dimension estimation using nearest neighbor graphs. *in: IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5418–5421, 2010.

[112] S.M. Steele, L.A. Shepp, and W.F. Eddy. On the number of leaves of a euclidean minimal spanning tree. *J. Applied Probability*, 24:809–826, 1987.

[113] F. Takens. On the numerical determination of the dimension of an attractor. In B.J. Braaksma, H.W. Broer, and F. Takens, editors, *Dynamical Systems and Bifurcations*, volume 1125 of *Lecture Notes in Mathematics*, pages 99–106. Springer Berlin Heidelberg, 1985.

[114] N. Tatti, T. Mielikainen, A. Gionis, and H. Mannila. What is the dimension of your binary data? *Proceedings of International Conference on data Mining*, 2006.

[115] J. Tenenbaum, V. Silva, and J. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2323, 2000.

[116] M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *J. of the Royal Statistical Society*, Series B, 61, Part 3:611–622, 1997.

[117] Jr. Tricot. Two definitions of fractional dimension. *Math. Proc. Cambridge Philos. Soc*, 91(1):57–74, 1982.

[118] G.V. Trunk. Statistical estimation of the intrinsic dimensionality of a noisy signal collection. *IEEE Trans. on Computers*, 25:165–171, 1976.

[119] M. Valle and A. R. Oganov. Crystal fingerprint space – a novel paradigm for studying crystal-structure sets. *Acta Crystallographica Section A*, 66(5):507–517, September 2010.

[120] V. Vapnik. *Statistical Learning Theory.* John Wiley and Sons, 1998.

[121] N. Verma. Distance Preserving Embeddings for General n-Dimensional Manifolds. *Journal of Machine Learning Research*, page 24152448, 2013.

[122] P. J. Verveer and R. P. W. Duin. An evaluation of intrinsic dimensionality estimators. *IEEE Trans. on PAMI*, 17:81–86, 1995.

[123] Q. Wang, S. Kulkarni, and S. Verdu. A nearest-neighbor approach to estimating divergence between continuous random vector. *Proc. ISIT*, pages 242–246, 2006.

[124] J. Wei, H. Peng, Y.S. Lin, Z.M. Huang, and J.B. Wang. Adaptive neighborhood selection for manifold learning. *Proceedings of International Conference on Machine Learning and Cybernetics*, pages 380–384, 2008.

[125] M. Wertheimer. Psycologische Forshung. Translation: A Source Book of Gestalt Psychology. 4:301–350, 1923.

[126] P.L. Zador. Ieee trans. information theory. *Asymptotic quantization error of continuous signals and the quantization dimension*, IT-28:139–148, 1982.

[127] P. Zhang, H. Qiao, and B. Zhang. An improved local tangent space alignment method for manifold learning. *Pattern Recognition Letters*, 32:181–189, 2011.

[128] Z. Zhang and H. Zha. Adaptive manifold learning. *Advances in Neural Information Processing Systems*, 17, 2005.

[129] H. Zou, T. Hastie, and R. Tibshirani. Sparse Principal Component Analysis. *Journal of Computational and Graphical Statistics*, 15:262–286, 2006.