UNIVERSITÀ DEGLI STUDI DI MILANO

Scuola di Dottorato in Matematica e Statistica
per le Scienze Computazionali - XXVII ciclo
Dipartimento di Matematica "Federigo Enriques"

Novel Techniques for Intrinsic Dimension Estimation

Relatore:
Prof. Paola Campadelli

Correlatore:
Dott. Elena Casiraghi

Coordinatore MaSSC:
Prof. Giovanni Naldi

Autore:
Claudio Ceruti

Anno Accademico 2013/2014

Contents

Introduction

Nowadays, dealing with massive amounts of data described by a huge number of characteristics is an everyday issue: computer scientist, physicists, economists, mathematicians, political scientists, bio-informaticists, sociologists, and many others, are gaining more and more access to huge collections of information. This is made possbile by new technologies that allows to collect and store data much more easily than in the past: in [48] it was estimated that during 2007, 2.9×10^{20} compressed bytes were stored, almost 2×10^{21} bytes were transmitted, and 6.4×10^{18} instructions per second were carried out on general-purpose computers, whereas general-purpose computing capacity grew at an annual rate of 58%, followed by the increase in globally stored information at 23%.

To better understand these numbers, some practical cases could come at hand: as an example,Wal-Mart Stores Inc. controls more than 1 million customer transactions every hour, which are then transferred into a database working with over 2.5 petabytes of information; the LHC experiments at Cern generated 40 terabytes every second; in bioinformatics one single microarray that measures gene expression is composed up to ten thousand of features; in climate analysis it's expected that the new systems for data collection would generate exabytes of information.

It is straightforward from these examples that the key issue today is how to deal with such an overabundant quantity of data, in other words, how to infer, extract and visualize meaningful information in an efficient and effective way. Even with techniques that can deal with these data in linear complexity in time and space, is still difficult to overcome the limitations that arise with an huge amount of data.

A viable way to handle these problems is to find a coherent representation of data, which has to be at the same time the most compact and the most informative possible.

One potential method to obtain this representation is to reduce the number of characteristics that describe the data, i.e. to reduce the data dimensionality. It is tempting to project the data to very few dimension, to two or three dimensions for the sake of visualization for example, but doing so could lead in information loss: having points sampled from the unit cube, that is points laying in the interval $[0, 1]^3$, and linearly projecting them to two dimensions will result in breaking the original spatial relationship between them (see figure 1).

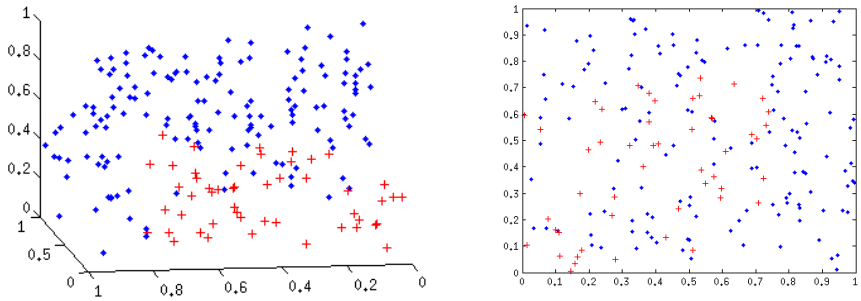

Figure 1: On the left: two well separated sets of points sampled from the unitary cube. On the right: the same sets linearly projected in two dimensions. The original spatial relationship between them is lost.

Even using a transformation that preserve the data dimension could wreck the data structure: taking points sampled from two bidimensional normal distribution with the same standard deviation but opposite means (i.e. μ and $-\mu$), and applying to them a simple function as the absolute value will result in the overlap of the points coming from different distributions (see figure 2).

The aforementioned examples tell us that we need some insight on the data structure to perform a data transformation that retains valuable information.

The first step in order to obtain a reliable description of the data structure is to determinate its dimension. To identify the dimension of the structure from the analysis of a given dataset $\mathbf{X}_N \equiv \{\mathbf{x}_i\}_{i=1}^N \subset \Re^D$, one useful information is the minimum number of parameters needed to represent the data without information loss, which is generally referred as `Intrinsic Dimensionality` (`i.d.`). To estimate the `i.d.` of $\mathbf{X}_N$ the data is generally viewed as composed by points constrained to lie on a low dimensional

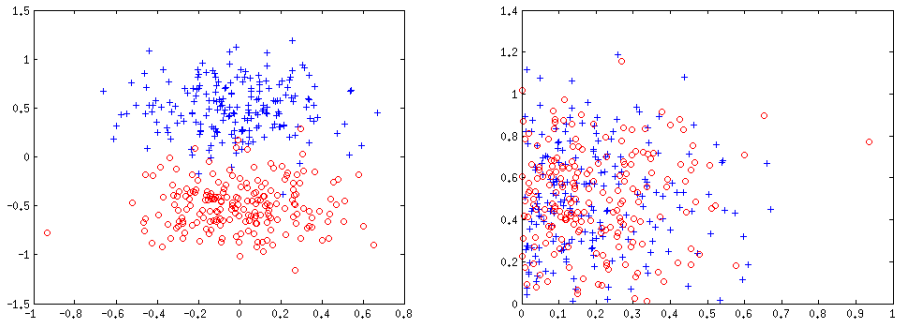

Figure 2: On the left: points sampled from two bidimensional normal distribution with opposite mean ($(0, 2)$ and $(0, -2)$) and same standard deviation. On the right: the same points transformed by means of the absolute value function.

smooth or locally smooth manifold[1] $\mathcal{M} \subset \Re^d$ embedded in a higher dimensional space $\Re^D$ by means of a linear or non linear map, where the dimensionality d of $\mathcal{M}$ is the `i.d.` value to be estimated. In more general terms, $\mathbf{X}_N$ is said to have `i.d.` equal to $d \in \{1, \cdots, D\}$ if its elements lie entirely within a d-dimensional subspace of $\Re^D$.

We can think at the `i.d.` as the minimum number of parameters needed to describe points sampled from the data structure, that is a lower bound on the dimension on which it is possible to project data without wrecking its structure. Therefore, to perform an accurate reduction of dimensionality, knowing the `Intrinsic Dimension` is a mandatory requirement.

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; this motivated a great deal of research effort devoted to the realization of reliable `i.d.` estimators, resulting in the large number of related techniques proposed in literature. However, despite all these studies, the state-of-the-art techniques for estimating the `i.d.` suffers from various limitations. For example, methods like the ones related to the Principal Component Analysis (`PCA`) assume that the embedding

[1] A topological space S is called locally Euclidean if there is a non-negative integer n such that every point in S has a neighborhood which is homeomorphic to the Euclidean space E^n. A topological manifold is a locally Euclidean second-countable Hausdorff space The dimension of a manifold is the dimension of the Euclidean space which every neighborhood is homeomorphic to; a manifold of dimension n is called a n-dimensional manifold or n-manifold.

of the data structure is done by means of a linear mapping. When these techniques are applied to a dataset which is composed by points sampled from a data structure which is non linearly embedded they generally produce an overestimate of the i.d.. Other estimators fail to provide reliable results when the i.d. is relatively high, i.e. when i.d.> 12. This is due to the fact that these techniques are based on statistic or geometric properties that can't describe the phenomena that occurs in high dimensional spaces, like norm compression, angle compression, empty space phenomena, and many others which will be described in Chapter 3.

The aim of this thesis is to present novel i.d. estimators that could provide reliable results overcoming the aforementioned limitations. The analysis of the drawbacks that affect the other methods gave us the bases for the creation of these i.d. estimators, called $\texttt{MiND}_{\texttt{ML}*}$,$\texttt{MiND}_{\texttt{KL}}$ and DANCo, that can deal whit high dimensional values of i.d. of data structures that are non linearly embedded.

We also note that there is no standard framework available to assess the performance and compare the results of the i.d. estimators. For this reason, we propose a new standard benchmark framework in order to objectively perform comparison of novel methods with the ones present in the literature. This framework is composed by synthetic and real publicly available datasets; we choose this particular set of datasets either because they are challenging or historically used in the i.d. estimation field. The proposed framework is equipped with methods and techniques in order to determine the quality of the estimators based on their results obtained on the datasets in the framework, along with robustness tests on the choice of the relative parameters and on the influence of the noise on the i.d. estimate. Methods for a statistical analysis and comparison of the estimators' results are also provided.

This thesis is organized as follows: in Chapter 1, notable state-of-the-art i.d. estimators will be surveyed, pointing out their advantages and limitations; in Chapter 2, the novel estimators $\texttt{MiND}_{\texttt{ML}*}$, $\texttt{MiND}_{\texttt{KL}}$ and DANCo will be described; in Chapter 3, a benchmark framework for experimental settings will be presented, and used to assess the quality of the estimators and compare their results; in Chapter 4, conclusions and open research problems will be outlined; in Appendix A, the pseudo-code of the estimators presented in Chapter 1 will be reported.

Chapter 1

State of the art

In this chapter we are going to survey the most interesting, widespread used, advanced state-of-the-art methodologies, pointing out the most important characteristics that depict them, and underlining their drawbacks and limitations.

In the first section some of the application domains in which the `i.d.` could be profitably used will be reported; they range from biology, to statistics, machine learning, physics, chemistry, genetics, finance, and many others.

After this, we are going to describe the state-of-the-art `i.d.` estimators, following a general categorization of the considered techniques. It's worth noting that is very complex to create a taxonomy in order to univocally group each method in different and well separated sets; this is due to the fact that in literature exists an huge collection of techniques, each of them sharing features with different methods. Hence, we choose five general categories, namely Projective, Fractal, Topological, Neighborhood-based, and Graph-based, which we believe underline the main ideas that inspired the development of the reported `i.d.` estimators.

Finally, for the sake of good order, a table summarizing all the described techniques, along with the corresponding categories, will be reported in the last part of the chapter.

1.1 Application Domains

In this section we motivate the increasing research interest aimed at the development of automatic `i.d.` estimators, and we recall different application

contexts where the knowledge of the i.d. of the available input datasets is a profitable information. As pointed out in the introduction, the i.d. is one of the first and fundamental information required by several dimensionality reduction techniques [105], which try to represent the data in a more compact, but still informative, way.

According to the statistical learning theory [106], the capacity and generalization capability of a given classifier may depend on the i.d.. 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 [106], pp. 156-158). Since the generalization error depends on the VC dimension, it follows that the generalization capability may depend on the i.d. value d. Moreover, in [39] 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 i.d. of the available dataset. When using an auto-associative neural network to perform a nonlinear feature extraction, the i.d. value d can suggest a reasonable value for the number of hidden neurons [59]. 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 [53] estimated on the dataset (see 8.6.2 of [5]), being d the number of hidden neurons. 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 i.d.s characterizing the considered object is an useful parameter in order to obtain a proper clustering [15].

In the field of gene expression analysis, the work proposed in [61] shows that the i.d. estimate computed by the nearest neighbor estimator [82] 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 crucial 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 [13], the authors show that i.d. estimation methods being derived from the basis theory of fractal dimensions ([41, 58, 65]), 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 [104] employs an `i.d.` estimator [41]; the experimental results show that `i.d.` is a useful information to be exploited when analyzing crystal structures. This study not only proves that `i.d.` 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 `i.d.` and embedded in spaces of much greater dimensionality.

The work of Carter [16] 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 `i.d.`. To separate the manifolds, the authors compute local `i.d.` estimates, by applying both a fractal dimension estimator and a nearest neighbor-based estimator on properly defined data neighborhoods. The authors then show that the computed local `i.d.`s might be helpful for the following interesting applications: (1) "Debiasing global `i.d.` estimates": the negative bias caused both by the limited number of available sample points and by the *curse of dimensionality* (see Chapter 2) is reduced by computing global `i.d.` 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 `i.d.`, since in high dimensions the points lye nearby the boundaries [3]. (2) "Statistical Manifold Learning": the local `i.d.` estimates are used to reduce the dimension of statistical manifolds [17], 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 `i.d.` 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 i.d.s).

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 [47] is to substitute state-of-the-art algorithms specifically designed to solve this task, with i.d. estimators. After motivating the idea by describing the relation between the i.d. of a dataset and the number of endmembers, the authors tested various i.d. estimators obtaining reliable results.

In [64], the authors use the information related to the i.d. to describe object representation in the anterior inferotemporal (AIT) cortex, based on responses of a large sample of cells stimulated with photographs of diverse objects. In this work, the authors reported that the dimensionality of AIT object representations is much lower than the dimensionality of the stimuli, and that various value of i.d. pertain to different representations in separated area of the visual system.

Finally, other noteworthy examples of research works that profitably exploit i.d., concern financial time series prediction [89], biomedical signal analysis [21, 76, 30], analysis of ecological time series [51], radar clutter identification [44], speech analysis [93], data mining and low dimensional representation of (biomedical) time series [50], plant traits representation [62].

1.2 Intrinsic Dimension Estimators

A taxonomy composed by well separated classes is not suitable to describe i.d. estimators by means of the main ideas they are based on: the high variety of methods present in literature, as well as the common features shared by many techniques, demands a "rough" subdivision based more on a bunch of general categories than on "hard" well separated classes. We selected the following categories, as the most representative and general ones:

- **Projective**:

  Projective estimators are the ones that explicitly compute the mapping that projects the input dataset $\mathbf{X}_N \subset \Re^D$ to the subspace $\mathcal{M} \subset \Re^d$ minimizing the information loss, and therefore view the i.d. as the

minimal number of vectors linearly spanning the subspace $\mathcal{M}$. It must be noted that projective techniques were originally designed for exploratory data analysis and dimensionality reduction, and generally required the `i.d.` as a parameter. The projective `i.d.` estimators arise from their extensions, that automatically calculate `i.d.`.

- **Neighborhood-based**:
  Methods that estimate the `i.d.` by means of data neighborhoods fall in this category. More specifically, these techniques describe data neighborhoods distribution as a function $f(d)$ of the `i.d.` d, usually assuming that close points could be modelled as uniformly drawn from small d-dimensional hyperspheres $\mathcal{B}_d(\mathbf{x}_c, r)$ having radius $r \to 0 \in \Re^+$ and being centered on $\mathbf{x}_c \in \mathcal{M}$.

- **Graph-based**:
  Techniques that exploit various types of graph structures for `i.d.` estimation have been proposed in literature. Among them, the most used structures are the `kNN` graph, the minimum spanning tree (`MST`) and its variation related to the geodesic, the geodesice minimum spanning tree (`GMST`), the sphere of influence graph (`SIG`), and its generalization, the k-sphere of influence graph (`kSIG`).

- **Fractal**:
  Fractal estimators are based on the assumption that the manifold $\mathcal{M}$ from which the data points are sampled, has a somehow fractal structure (see [33] for an exhaustive description of fractal sets). Even if the `i.d.` estimators defined as fractal are based on different definitions of the dimension of a fractal structure, they share the basis concept that the volume of a d-dimensional ball of radius r scales with its size as r^d [33, 97]. According to this, all fractal dimension estimators are based on the idea of counting the number of observations in a neighborhood of radius r to estimate the rate of growth of this number. If the estimated growth is r^d , then the estimated `i.d.` of the data is considered to be equal to d.

- **Topologic**:
  Estimators that are defined as topological consider the `i.d.` to be estimated as equivalent to the topological dimension of the manifold

or the equivalent Lebesque's Covering Dimension, defined by means of topological covering and their refinement. More formally, the topological dimension of the topological space $\mathcal{X}$, is d if every finite cover[1] of $\mathcal{X}$ admits a refinement $\mathcal{C}'$ such that no subset of $\mathcal{X}$ has more than $d+1$ intersecting open sets in $\mathcal{C}'$. If no such minimal integer value exists, $\mathcal{X}$ is said to be of infinite topological dimension

For the last category, the topological one, it is worth noting that there is some ambiguity in the use of this term in literature: estimators that provide an integer value for the `i.d.` are marked as topological, due to the fact that they provide an `i.d.` value that is considered equivalent to the topological dimension of the manifold. In this work we follow a more strict convention, in which we consider as topological only the `i.d.` estimators that are based on the construction of a covering, or an approximation of it.

1.2.1 Projective Estimators

Among the projective `i.d.` estimators, `PCA` [53, 68] is one of the most cited and well known technique for exploratory data analysis and linear dimensionality reduction, 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 `i.d.` estimation, the estimate is the number of "most relevant" eigenvectors of the sample covariance matrix, also called principal components (`PC`s). 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` [90]), the local `PCA` (`LPCA` [40]) and its extensions to automatically select the number of `PC`s [107, 12]. We observe that the work presented in [12] is one of the first that estimates `i.d.` by considering an underlying topological structure, and therefore applies `LPCA` on data neighborhoods. 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 [65].

[1] Given a topological space $\mathcal{X}$, a cover set $\mathcal{Y} \subseteq \mathcal{X}$ is a countable collection $\mathcal{C} = \{\mathcal{C}_i\}$ of open sets such that each $\mathcal{C}_i \subset \mathcal{X}$ and $\cup_i \mathcal{C}_i \supseteq \mathcal{Y}$. A refinement of a cover $\mathcal{C}$ of a set $\mathcal{Y}$ is another cover $\mathcal{C}'$ such that each set in $\mathcal{C}'$ is contained in some sets of $\mathcal{C}$.

The usage of a probabilistic approach has been firstly introduced by Tipping and Bishop in [99]. 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 i.d.-estimation mechanism depending on critical thresholds. This motivates its subsequent variants [32] and developments, whose examples are Bayesian PCA (BPCA [6]), and two Bayesian model order selection methods introduced in [86, 77]. In [8] the asymptotic consistency of i.d. 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 i.d. estimation results have shown to be unreliable [58, 65], in the following recent and promising proposals are described with more details.

The Simple Exponential Family PCA (SePCA [67]) 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 [42] the authors propose the Sparse Probability PCA (SPPCA) as a probabilistic version of the Sparse PCA (SPCA [112]). Precisely, SPCA selects i.d. 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 ([7]-Section 3.5). The authors' results and also the results of the comparative evaluation proposed in [18] show that this method seems to be less affected by the problems of the aforementioned techniques.

An alternative method (MLSVD, [69]) that applies Singular Value Decomposition (SVD) instead of PCA, locally and in a multi-scale fashion to estimate the i.d. characterizing D-dimensional datasets drawn from non-linearly

embedded d-dimensional manifolds $\mathcal{M}$ corrupted by Gaussian noise. Precisely, exploiting the same ideas of the theoretical PCA-based i.d. estimator presented in [57], 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 $\mathcal{B}(\mathbf{x}, r)$ centered on the data points $\mathbf{x}$ 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 $\mathcal{M} \cap \mathcal{B}$ 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 $T^d_{\mathcal{M}}(\mathbf{x}, r)$, computed by applying SVD on the k neighbors, is a good approximation of the tangent space of $\mathcal{M} \cap \mathcal{B}$ and the number of its relevant eigenvalues correspond to the (local) i.d. of $\mathcal{M}$. To find a proper value for r, the authors propose a multi-scale approach that applies SVD on neighborhoods $\mathcal{B}(\mathbf{x}, r_s)$ whose radius varies in a range $r_s \in \{r_L..r_H\}$. This allows to compute D scale-dependent, local singular values $\lambda_1(\mathbf{x}, r_s) \geq \ldots \geq \lambda_D(\mathbf{x}, r_s)$; 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 i.d. estimate $\hat{d}$ that maximizes the gap $\Delta(j) = \lambda_j(\mathbf{x}, r_s) - \lambda_{j+1}(\mathbf{x}, r_s)$ 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 well known methods [41, 65, 27, 45, 43, 20, 16] show that the i.d.s estimated on the same datasets are unreliable also in the absence of noise.

1.2.2 Fractal Estimators

The first and one of the most cited, especially for historical reasons, fractal estimator of the i.d. is presented in [41]. It is an estimator of the **Correlation Dimension** (dim_{Corr}), and will be referred as CD in the following; the formal definition of Correlation Dimension is based on the correlation integral defined as, given a finite sample set $\mathbf{X}_N$:

$$C_N(r) = \frac{2}{N(N-1)} \sum_{i=1, i<j}^{N} I(r - \|\mathbf{x_i} - \mathbf{x_j}\|)$$

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

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

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 `i.d.` estimates, this estimator needs a very large number of data points [104], which is never available for practical applications; however the computed unreliable estimations can be corrected by the correction method proposed in [14].

The relevance of the `CD` estimator is shown by its several variants and extensions. An example is the work proposed in [97], 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 [96] the authors estimate the `i.d.` 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 [2] the authors choose a different definition of the fractal dimension, namely the **Information Dimension** dim_I defined as:

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

where $\mathcal{N}(\delta)$ 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 the above equation is big enough the different coverings used to estimate dim_I could produce different values for $\mathcal{N}(\delta)$,

the author look for the covering composed by the minimum number $\mathcal{N}_{min}(\delta)$ of nonempty sets. Similar to the `CD` algorithm, the estimated `i.d.` based on dim_I is the average slope of the curve obtained by fitting the points with coordinates $\left(\log\ \delta;\ \sum_{i=1}^{\mathcal{N}_{min}(\delta)}\ pr_i \log\ pr_i\right)$.

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 correct value of `i.d.` if the data distribution on the manifold is nearly non-uniform, in [58] the author proposes the Packing Number (`PN`), an `i.d.` estimator that approximates the **Capacity Dimension** (dim_{Cap}). To formally define dim_{Cap}, the ϵ-covering number $\mathcal{N}(\epsilon)$ of a set $\mathcal{S} \subset \mathcal{X}$ must be defined; $\mathcal{N}(\epsilon)$ is the minimum number of open balls $\mathcal{B}(\mathbf{x}_0, \epsilon) = \{\mathbf{x} \in \mathcal{X} :\ \|\mathbf{x}_0 - \mathbf{x}\| < \epsilon\}$ whose union is a covering of $\mathcal{S}$, where $\|\cdot\|$ is a distance metric. The definition of dim_{Cap} of $\mathcal{S} \subset \mathcal{X}$ is based on the observation that the covering number $\mathcal{N}(\epsilon)$ of a d-dimensional set is proportional to ϵ^{-d}:

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

Since the estimation of $\mathcal{N}(\epsilon)$ is computationally expensive, based on the relation $\mathcal{N}(\epsilon) \leq \mathcal{N}_{pack}(\epsilon) \leq \mathcal{N}(\frac{\epsilon}{2})$, the authors employ the ϵ-Packing number $\mathcal{N}_{pack}(\epsilon)$, defined in [100] as the maximum cardinality of an ϵ-separated set. Employing a greedy algorithm to compute $\mathcal{N}_{pack}(\epsilon)$, the estimate, $\hat{d}$, of dim_{Cap} is computed as:

$$\hat{d}\,(\epsilon_1, \epsilon_2) = -\frac{\log \mathcal{N}_{pack}(\epsilon_1) - \log \mathcal{N}_{pack}(\epsilon_2)}{\log \epsilon_1 - \log \epsilon_2}$$

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 `i.d.` 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 [45] the authors propose an estimator (`Hein`) based on the asymptotes of a smoothed version of the correlation

integral, obtained by replacing the step function $I(\cdot)$ with a suitable kernel function. Precisely, they define:

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

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 the above equation, 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 `i.d.` 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 $n_{sub} \in \mathcal{N}_{sub} = \{N, N/2, N/3, N/4, N/5\}$ and the D curves whose points have coordinates $(U(n_{sub}, h(n_{sub}), d), n_{sub})$ are considered. Employing this information the following `i.d.` estimator is proposed:

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

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, 18].

In [85] the authors present an estimator derived by the analysis of a vector quantizer applied to datasets $\mathbf{X}_N \subseteq \Re^D$. Considering the codebook $\mathcal{Y} = \{\mathbf{y}_1..\mathbf{y}_k\} \subset \Re^D$ containing k code-vectors $\mathbf{y}_i$, a *k-point quantizer* is defined by a measurable function $Q_k : \Re^D \to \mathcal{Y}$, which brings each data point to one of the code-vectors in $\mathcal{Y}$. This partitions the dataset into k so-called *quantizer cells* $\mathcal{S}_i = \{\mathbf{x}_i \in \mathbf{X}_N : Q_k(\mathbf{x}_i) = \mathbf{y}_i\}$, where $\log_2(k)$ is called the *rate of the quantizer*. Being $\mathbf{X}$ a random vector distributed according to a probability distribution ν, the *quantization error* is $e_r(Q_k|\nu) = (E_\nu[\|\mathbf{X} - Q_k(\mathbf{X})\|^r])^{\frac{1}{r}}$, where $r \in [1,\infty)$ and $\|\cdot\|$ is the Euclidean norm in $\Re^D$. Given the set $\mathcal{Q}_k$ of all D-dimensional k-point quantizers, the performance achieved by an optimal k-point quantizer on $\mathbf{X}$, is

$e_r^*(Q_k|\nu) = \inf_{Q_k \in \mathcal{Q}_k}(e_r(Q_k|\nu))$. When the quantizer rate is high, the quantizer cells can be well approximated by D-dimensional hyperspheres with radius equal to ϵ and centered on each code-vector $\mathbf{y}_i \in \mathcal{Y}$. 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 [111] 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 [111] confirms that, for a regular ν supported on the manifold $\mathcal{M}$, $d_r(\nu)$ exists for each $1 \leq r \leq \infty$ and equals the `i.d.` of $\mathcal{M}$. 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 [111, 54], 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 `i.d.` 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, in [16] this statement is confuted with theoretical arguments.

It must be underlined that all the derived estimators described so far have the fundamental limitation that in order to get an accurate estimation, the size N of the dataset with `i.d.` d has to satisfy the inequality proved by Eckmann in [31] for the `CD` estimator:

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

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

1.2.3 Topological Estimators

To our knowledge, at the state-of-the-art only two estimators have been explicitly designed to estimate the Topological Dimension: the Tensor Voting Framework (`TVF`, [75]) and the method presented in [66].

`TVF` and its variants [70] relies on the usage of an iterative information diffusion process based on Gestalt principles of perceptual organization [110]. `TVF` iteratively diffuses local information describing, for each $\mathbf{x}_i \in \mathbf{X}_N$, the tangent space approximating the underlying neighborhood of $\mathcal{M}$. 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 $\mathbf{T}_i^0$, which is an identity matrix representing the absence of orientation, is used to initialize a token x_i for each point $\mathbf{x}_i$ as $\{x_i = (\mathbf{x}_i, \mathbf{T}_i^0)\}_{i=1}^N$. During iteration t each token x_i "generate" the set of tensors $\left\{\mathbf{T}_{i,j}^t\right\}_{j \neq i}$ that enact as votes cast to neighboring tokens; precisely, $\mathbf{T}_{i,j}^t$ is sent to the j^{th} neighbor, it encodes informations related to the local tangent space estimate in $\mathbf{x}_i$ 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 x_j receives votes that are summed to update the x_j's tensor as $\mathbf{T}_j^{t+1} = \sum_{i \neq j} \mathbf{T}_{i,j}^t$; this essentially refines the estimate of the local tangent space in $\mathbf{x}_j$. `TVF` can be employed to estimate the local `i.d.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 [87] that the topological dimension of any $\mathcal{M} \subseteq \Re^d$ coincides with the affine dimension d of a finite simplicial complex[2] covering $\mathcal{M}$. 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 $\mathcal{M}$ could be practically estimated by analyzing the number of vertices of the collection of simplexes estimated on $\mathbf{X}_N$.

To this aim, in [66] a method is proposed to find the number of relevant positive coefficients that are needed to reconstruct each $\mathbf{x}_i \in \mathbf{X}_N$ from a linear combination of its k neighbors, where k is a parameter to be manu-

[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.

ally 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 dimension of $\mathcal{M}$. 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 `i.d.` 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 $\bar{\mathbf{c}}$ of the sorted coefficients, and estimates `i.d.` as the number of relevant values in $\bar{\mathbf{c}}$. Note that, since $k > d$, this method is strongly affected by the curvature of the manifold when the `i.d.` 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 `i.d.` (`i.d.`≤ 6), under the assumption that the sampling process is uniform and the data points are sufficient.

As well as `TVF`, this approach has shown to be effective only for manifolds of low curvature as well as low `i.d.` values.

1.2.4 Neighborhood-based Estimators

As one of the first estimator to exploits neighborhood informations, Trunk's method [101] is often cited. 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 $\mathbf{x}_i \in \mathbf{X}_N$, 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 `i.d.` 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.

With the same base idea, the work presented by Pettis [82] is notable since it is one of the first providing a mathematical motivation for the use of nearest-neighbor distances.

To this end, the authors consider that the probability $p(x \in \mathcal{B}_d(\mathbf{x}_0, r))$ of a point $\mathbf{x}$ to be in the d-dimensional hypersphere $\mathcal{B}_d(\mathbf{x}_0, r)$ with radius r

and centered on a point $\mathbf{x}_0$ is:

$$p(\mathbf{x} \in \mathcal{B}_d(\mathbf{x}_0, r)) = r^d \int_{z \in \mathcal{B}_d(0,1)} g(rz + \mathbf{x}_0) d\sigma(z) = r^d \mu(\mathbf{x}_0, r)$$

where $g(\cdot)$ is the point distribution function, and σ is the Lebesgue measure of $\mathcal{M}$ defined from its volume form. For small values on r, $\mu(\mathbf{x}_0, r)$ could be considered a constant μ_0, thus obtaining:

$$p(\mathbf{x} \in \mathcal{B}_d(\mathbf{x}_0, r)) = r^d \mu_0 \Rightarrow \log(p(\mathbf{x} \in \mathcal{B}_d(\mathbf{x}_0, r))) = d \log(r) + \log(\mu_0)$$

Exploiting this relation, local `i.d.` estimates $\hat{d}_i$ are computed for each $\mathbf{x}_i \in \mathbf{X}_N$, and the results are then averaged to get the global `i.d.` estimate d. More precisely, considering each $\mathbf{x}_i \in \mathbf{X}_N$ as the center of a ball $\mathcal{B}_d(\mathbf{x}_i, r)$, the authors compute the distance r_i between $\mathbf{x}_i$ and its k th-nearest neighbor (being k a parameter to be fixed). Then, since $p(\mathbf{x} \in \mathcal{B}_d(\mathbf{x}_i, r_i^{(k)})) \simeq \frac{k}{N}$ and similarly, by considering a number $\frac{k}{2}$ of nearest neighbors, $p(\mathbf{x} \in \mathcal{B}_d(\mathbf{x}_i, r_i^{(\frac{k}{2})})) \simeq \frac{k}{2N}$, the relations $\log(\frac{k}{N}) \simeq d \log(r_i^{(k)}) + \log(\mu_0)$ and $\log(\frac{k}{2N}) \simeq d \log(r_i^{(\frac{k}{2})}) + \log(\mu_0)$ can be used to deduce $log(2) \simeq d \log\left(\frac{r_i^k}{r_i^{\frac{k}{2}}}\right)$, so that a local estimate $\hat{d}_i$ is computed as

$$\hat{d}_i = \frac{\log(2)}{\log\left(\frac{r_i^k}{r_i^{\frac{k}{2}}}\right)}$$

Therefore, the global `i.d.` estimate $\hat{d}$ is:

$$\hat{d} = \frac{N \log(2)}{\sum_{i=1}^{N} \left(\frac{r_i^k}{r_i^{\frac{k}{2}}}\right)}$$

Since this algorithm is limited by the choice of a suitable value for parameter k, in [107] 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 `i.d.` when its value is high.

Taking into account the relation

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

in [34] the number $N_{\mathcal{B}_d}$ of data points in $\mathcal{B}_d(\mathbf{x}, r)$ is described by a polynomial $f(r) = \sum_{s=0}^{d} \beta_s r^s$ of degree d. In practice, considering $\mathbf{x}_i, \mathbf{x}_k \in \mathbf{X}_N$, calling $r_{ik} = \|\mathbf{x}_i - \mathbf{x}_k\|$ the inter-point distances, and being $r = \min_{i,k=1}^{N} r_{ik}$, and R a parameter adaptively estimated, a set of n radius values $\mathbf{r} = \left\{ r_j = r + \frac{j(R-r)}{n} \right\}_{j=1}^{n}$ 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 [39] is used to estimate the degree $\hat{d}$ of $\hat{f}$, which is considered as the `i.d.` estimate. Maximum Likelihood Estimator, `MLE` [65], one of the most cited estimators, treats the neighbors of each point $\mathbf{x}_i \in \mathbf{X}_N$ as events in a Poisson process and the distance $r^{(j)}(\mathbf{x}_i)$ between the query point $\mathbf{x}_i$ and its j^{th} nearest neighbor as the event's arrival time. Since this process depends on d, `MLE` estimates `i.d.` by maximizing the log-likelihood of the observed process. In practice a local `i.d.` estimate is computed as:

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

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

The theoretical stability of the proposed `i.d.` 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 [81, 4]. Though the authors' comparative evaluation shows the superior performance of the proposed estimator w.r.t. the `CD` estimator [41] and the `NN` estimator [82], 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 `i.d.` 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 [73], where the arithmetic mean is substituted with the harmonic average, leading to the following estimator: $\hat{d}(k) = \left(\frac{1}{N} \sum_{i=1}^{N} \frac{1}{\hat{d}(\mathbf{x}_i, k)} \right)^{-1}$.

Though the proposal in [73] seems to achieve more accurate results, it is based on the assumption that neighbors surrounding each $\mathbf{x}_i$ are independent, which is clearly incorrect. To cope with this problem, in [29] an

interesting regularized version of MLE applies a regularized maximum likelihood technique to distances between neighbors. The comparative evaluation with the aforementioned MLE methods [65, 73] 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 [56, 55] a further improvement of MLE is presented; it achieves a better performance by substituting euclidean distances with geodesic ones.

1.2.5 Graph-based Estimators

As noted in [11], the work of [84] has cleared that theories underlying graphs can be applied to solve a variety of statistical problems; indeed, also in the field of i.d. estimation various types of graph structures have been proposed [11, 46, 26, 25] and used for i.d. estimation. Among them, the most used structures are the kNN graph, the minimum spanning tree (MST) and its variation related to the geodesic, the geodesice minimum spanning tree (GMST), the sphere of influence graph (SIG), and its generalization, the k-sphere of influence graph (kSIG). More precisely, a $\texttt{kNN}(\mathbf{X}_N)$ is built employing a distance function to weight the arcs connecting each $\mathbf{x}_i$ to its kNNs. A $\texttt{MST}(\mathbf{X}_N)$ is the spanning tree minimizing the sum of the edge weights. When the weights approximate geodesic distances, a $\texttt{GMST}(\mathbf{X}_N)$ is obtained. A $\texttt{SIG}(\mathbf{X}_N)$ is defined by connecting nodes $\mathbf{x}_i$ and $\mathbf{x}_j$ iff $\|\mathbf{x}_i - \mathbf{x}_j\| \leq \rho(i) + \rho(j)$, where $\rho(i)$ is the distance between $\mathbf{x}_i$ and its nearest neighbor in $\mathbf{X}_N$; in other words, two vertices are connected if their corresponding neighborhood hyperspheres intersect. The generalization $\texttt{kSIG}(\mathbf{X}_N)$ of $\texttt{SIG}(\mathbf{X}_N)$ use as the intersecting hyperspheres the ones generated by kNN In [26, 27], after defining the length functional $\mathcal{L}(G_N(\mathbf{X}_N)) = \sum |e_{i,j}|^\gamma$, $\gamma \in (0, d)$, to build either the $\texttt{GMST}(\mathbf{X}_N)$ or the $\texttt{MST}(\mathbf{X}_N)$ of $\texttt{kNNG}(\mathbf{X}_N)$, graph theories are exploited to estimate both the i.d. of the underlying manifold structure $\mathcal{M}$ and its intrinsic Rènyi α-entropy $\mathcal{H}_\mathcal{M}$. To this aim, the authors derive the linear model: $\log \mathcal{L}(\texttt{MST}(\mathbf{P}_N)) = a \log d + b,\ a = (d - \gamma)/d,\ b = \log c + \mathcal{H}_\mathcal{M}$, being c an unknown constant, and exploit it to define an estimator of both d and $\mathcal{H}_\mathcal{M}$. Briefly, a set of cardinalities $\{n_k\}_{k=1}^K$ is chosen and, for each n_k, the $\texttt{MST}(\mathbf{X}_{n_k})$ is constructed on the set $\mathbf{X}_{n_k}$, which contains n_k points randomly sampled from $\mathbf{X}_N$, to obtain a set of K pairs $(\log \mathcal{L}(\texttt{MST}(\mathbf{X}_{n_k})), n_k)$. 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 i.d. is calculated as $\hat{d} = \text{round}\{\gamma/(1 - \hat{a})\} \simeq d$. This process is iterated to produce the final estimate as the average of the obtained results.

The aforementioned kNNG based algorithm [27, 26] is exploited in [25], where the authors consider data sets sampled from a union of disjoint manifolds with possibly different i.d.s. To estimate the local i.d.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 i.d.s, and their sampling distributions.

In [11] the authors present three i.d. 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 $S_N^1(\mathbf{X}_N) = \bar{r}_j(\texttt{kNNG}(\mathbf{X}_N))$ in the following, is based on the reach[3] of vertices in the $\texttt{kNNG}(\mathbf{X}_N)$. Considering that the reach of each vertex $\mathbf{x}_i \in \texttt{kNNG}(\mathbf{X}_N)$ grows as the i.d. increases, in [10] the average reach $\bar{r}_j(\texttt{kNNG})$ in j steps of vertices in $\texttt{kNNG}(\mathbf{X}_N)$ is employed: $S_N^1(\mathbf{X}_N) = \bar{r}_j(\texttt{kNNG}(\mathbf{X}_N)) = \frac{1}{N}\sum_{i=1}^{N} r_{j,i}(\mathbf{x}_i, \texttt{kNNG}(\mathbf{X}_N))$.

The second statistic, denoted with $S_N^2(\mathbf{X}_N) = M_N(\texttt{MST}(\mathbf{X}_N))$, is computed by considering the degree of vertices in the $\texttt{MST}(\mathbf{X}_N)$. Recalling that, for datasets $\mathbf{X}_N$ obtained from a continuous distribution on $\Re^d$, the ratio of nodes with a given degree j in $\texttt{MST}_N(\mathbf{X}_N)$ converges a.s. to a limit depending only on j and d [95], 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 i.d.. This leads to the definition of an i.d. estimator employing the statistic $S_N^2 = M_N(\texttt{MST}(\mathbf{X}_N)) = \frac{1}{N}\sum_{i=1}^{N}(deg_{MST(\mathbf{X}_N)}(\mathbf{x}_i))^2$.

The third statistic, denoted as $S_N^3(\mathbf{X}_N) = U_N^k(\texttt{kSIG}(\mathbf{X}_N))$, is motivated by studies in the literature [91] showing that the expected number of neighbors shared by a given pair of points depends on the i.d. of the underlying manifold. Accordingly, calling $N_{i,j}$ the number of samples in the intersection of the two kNN hyperspheres centered on $\mathbf{x}_i$ and $\mathbf{x}_j$, intuitions similar to those considered for $\bar{r}_j(\texttt{kNNG})$ lead to define $S_N^3(\mathbf{X}_N) = U_N^k(\texttt{kSIG}(\mathbf{X}_N)) =$

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

$\frac{1}{n}\sum_{i\leq j} N_{i,j}$.

Based on their theoretical results and empirical tests on synthetically generated datasets characterized by `i.d.` values d_j in a finite range $\mathbf{F} \subseteq N^+$ (where $\mathbf{F} = \{d_j\}_{d_j=2}^{12}$ in the reported experiments), the authors propose an approximate Bayesian estimator that could indistinctly employ each of the three statistics S_N^1, S_N^2, and S_N^3, 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 $d_j \in \mathbf{F}$, 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 `i.d.` $= d_j$, of the Gaussian density $f_{d_j}(\cdot)$ of S_N^*.

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

$$P[d_j|S_N^*] = \frac{\tilde{f}_{d_j}(s_{\mathbf{X}})}{\sum_{d_j \in F} \tilde{f}_{d_j}(s_{\mathbf{X}})},\ d_j \in \mathbf{F}$$

and employed to compute an "a posteriori expected value" of the `i.d.`:

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

The authors evaluate the performance of their methods on synthetic datasets, some of which have been used by similar studies in literature [45], 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 [65, 35, 25, 94] 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.

1.3 Summary

The `i.d.` estimators described so far are listed in table 1.1, along with their relative categories; we also inserted in the last two columns a different

taxonomy which was commonly used by several authors in the past. It viewed methods as global, when `i.d.` 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. We can note that the majority of the recent methods have abandoned the global approach since the analysis of a dataset at its biggest scale could produce unreliable results. as we'll explain in the next chapter. In the next chapter we are also going to present novel estimators of the `i.d.`, precisely $\texttt{MiND}_{\texttt{ML}*}$,$\texttt{MiND}_{\texttt{KL}}$, and `DANCo`, which are local neighborhood-based methods.

	Projective	**Topologic**	**Fractal**	**Neighborhood-based**	**Graph-based**	**Global**	**Local**
PCA [53, 68]	⋆					⋆	
KPCA [90]	⋆					⋆	
LPCA [40]	⋆						⋆
PPCA [99]	⋆					⋆	
BPCA [6]	⋆					⋆	
SePCA [67]	⋆					⋆	
SPPCA [42]	⋆					⋆	
MLSVD [69]	⋆						⋆
CD [41]			⋆				⋆
INFOdim [2]			⋆				⋆
PN [58]			⋆				⋆
Hein [45]			⋆				⋆
Quantization Dimension [85]			⋆				⋆
TVF [75]		⋆					⋆
Simplicial based [66]		⋆					⋆
Trunk [101]				⋆			⋆
NN [82]			⋆				⋆
Polynomial [34]				⋆			⋆
MLE [65]			⋆	⋆			⋆
GMST [26, 27, 25]					⋆	⋆	
kNNG [26, 27, 25]					⋆	⋆	

Table 1.1: `i.d.` estimation techniques reported in section §1.2 along with their relative categories

Chapter 2

Novel intrinsic dimension estimators

In the previous chapter we have presented state-of-the-art estimators of the `i.d.`, as well as their drawbacks and limitations. We are now going to describe novel techniques that overcomes the shortcomings that affects the presented methods.

In the first section of this chapter we are going to discuss a local approach for `i.d.` estimators, named $\texttt{MiND}_{\texttt{ML}*}$ (`Minimum Neighbor Distance` based on `Maximum Likelihood`), that are based on the maximum likelihood of distributions related to neighborhood distances.

After underling the limitations of these estimators when confronted with high dimensional spaces, we will introduce a `pdf` comparison approach to overcome the flaws of the previous techniques. More precisely, we will present $\texttt{MiND}_{\texttt{KL}}$ [88, 71] which uses a Kullback-Leibler divergence between distance-related `pdf`s estimated locally on the dataset and on synthetic generated data of various dimensions ($d \in \{1, \cdots, D\}$).

In the last section `DANCo` [18] will be introduced; this `i.d.` estimator improves $\texttt{MiND}_{\texttt{KL}}$ through the addition of angle-related information, obtaining even more precise results. An efficient version of `DANCo` (called `FastDANCo`) is also presented, together with a more feasible technique for the construction of the `kNN`.

2.1 A Local Approach Model

In the previous chapter we described various techniques for `i.d.` estimation. Among them, the more accurate and reliable resulted to be the ones based on a local analysis of data. These methods, in order to provide an `i.d.` estimate, exploit the local properties of a small neighborhood of a point of the dataset, under the assumption that these properties are the ones that characterize the manifold from which the data are sampled. More specifically, consider a manifold $\mathcal{M} \equiv \Re^d$ embedded in an higher dimensional space $\Re^D$ through a locally isometric non-linear smooth map $\psi : \Re^d \to \Re^D$, from which the data points are sampled by means of a smooth non-uniform sampling `pdf` $f : \mathcal{M} \to \Re^+$; with a local approach for `i.d.` estimation we are willing to find local manifold proprieties depending only on d, that could be calculated by looking at the neighborhood of finite sets of data points in $\Re^D$. This requires a model of the data points neighborhood that is representative of the corresponding point neighborhood on the manifold, according to the property of a manifold to be locally euclidean. In this work, in order to estimate manifold properties by means of local information, we model each neighborhood of a point in the dataset as a set of points uniformly sampled from a d-dimensional hypersphere, where d is the dimension of the manifold, having radius equal to the neighborhood size, and centred on the given point. The first problem of this model arises when considering that the sampling distribution by means of which the data points are extracted from the manifold, i.e. $f : \mathcal{M} \to \Re^+$, is generally not uniform; this is the opposite of our model assumption that the points are uniformly sampled inside the hypersphere. The following theorem states that this assumption is indeed correct, when considering neighborhoods of very small size. First of all, we define, without loss of generality, the center of the hypersphere to be in $\mathbf{0}_d$, and we show that any smooth `pdf` f is locally uniform where the probability is not zero. To this aim, assuming $f(\mathbf{0}_d) > 0$ and $\mathbf{z} \in \Re^d$, we denote with f_ϵ the `pdf` obtained by setting $f_\epsilon(\mathbf{z}) = 0$ when $\|\mathbf{z}\| > 1$, and $f_\epsilon(\mathbf{z}) \propto f(\epsilon\mathbf{z})$ when $\|\mathbf{z}\| \leq 1$. More precisely, denoting with $\chi_{\mathcal{B}_d(\mathbf{0}_d,1)}$ the indicator function on the ball $\mathcal{B}_d(\mathbf{0}_d, 1)$, we obtain:

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

Theorem 2.1. *Given* $\{\epsilon_i\} \to 0^+$, $f_\epsilon(\mathbf{z})$ *describes a sequence of* ***pdfs*** *having*

the unit d-dimensional ball as support; such sequence converges uniformly to the uniform distribution $\mathbf{B}_d$ *in the ball* $\mathcal{B}_d(\mathbf{0}_d, 1)$.

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

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

Defining:

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

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

$$\begin{aligned}
V \cdot min(\epsilon) \leq &\quad Vf(\epsilon\mathbf{z}) &&\leq V \cdot max(\epsilon) \\
V \cdot min(\epsilon) \leq &\quad \textstyle\int_{\mathcal{B}_d(\mathbf{0},1)} f(\epsilon\mathbf{t})d\mathbf{t} &&\leq V \cdot max(\epsilon)
\end{aligned}$$

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

This theorem states that if a neighborhood of a point in the dataset is small enough, we can think at it as a representative of a neighborhood of the same point on the manifold.

The problem is that having a finite set of points we can not guarantee that the hypersphere model can describe the neighborhood. The base technique for selecting the neighbors of a given point is the `k-Nearest Neighbor`, which simple selects the nearest k points by means of Euclidean distance. The value of k, that is the number of neighbors, is a parameter that has to be choose very carefully. First of all, the above theorem, as well as theorem 1 and theorem 4 in [27] guarantees that the hypersphere model is true only when $k \to \infty$, and this requirement is often translated into practice by several `i.d.` estimators by requiring the number k of available empirical neighboring samples to be sufficiently high. Taking as much points

as possible is not a good way to fulfil this requirement, due to the fact that in most practical case this means that the neighborhood's size is very large compared to the manifold diameter, that could results in an incorrect `i.d.` estimate (see figure 2.1)

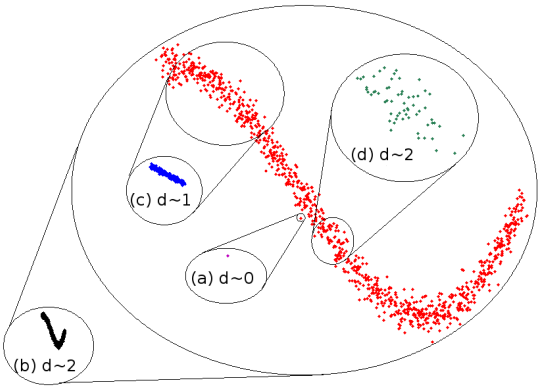

Figure 2.1: At very small scales (a) the dataset seems zero-dimensional; in this example, when the resolution is increased until including all the dataset (b) the `i.d.` looks larger and seems to equal the embedding space dimension; the same effect happens when it is estimated at noise level (d); the correct `i.d.` estimate is obtained at an intermediate resolution

Even selecting a neighborhood with a correct size could lead in improper results: as the numbers of neighbors k increases, the probability of selecting points which are not neighbors on the manifold's surface grows correspondly (see figure 2.2).

To better explain this, we informally define a geodesic path as the minimum path that connects two manifold's points and lies entirely on the manifold's structure, and we define an euclidean path as the minimum path measured in terms of euclidean distance. The increment of the neighborhood's radius will often lead into a discrepancy between the geodesic path and the euclidean path, resulting in neighborhood that are no longer representative of the manifold's structure: the theorem 4 in [27] proves that geodetic paths converge to euclidean paths with probability 1 only in the infinitesimal neighborhood.

Assuming that the neighborhood is properly chosen, we are now going to describe the hypersphere model in details. Our aim is to exploit this model

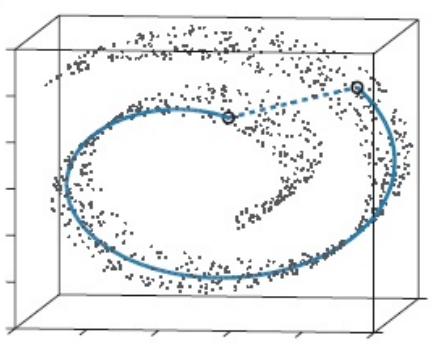

Figure 2.2: The solid line is the geodesic path between two points on the Swiss Roll dataset. The dashed line is the euclidean distance between the considered points. The discrepancy between the geodesic path and the euclidean path could lead in the construction of neighborhoods which are not representative of the structure of the manifold.

to infer an estimate for the `i.d.`; we are going to do so by means of the probabilistic density function relative to the distances of points uniformly sampled on the hypersphere from its center. More formally, considering the unit hypershpere $\mathcal{B}_d(\mathbf{0}_d, 1) \subset \Re^d$ centerd in the origin and k points $\{\mathbf{z}_i\}_{i=1}^k$ uniformly drawn from it, our aim is to find the `pdf` related to the minimum distance between the k points and $\mathbf{0}_d$. Call $p(r)$ the `pdf` for the event $\|\mathbf{z}_i\| = r \quad (r \in [0, 1])$, where $\|\cdot\|$ is the L_2 norm operator, and denote with $P(\tilde{r} < r)$ the probability for the event $\|\mathbf{z}_i\| < r$; being $\mathbf{z}_i$ uniformly drawn it is possible to evaluate these probabilities by means of hypersphere volume ratios. The volume of a d dimensional hypersphere of radius r is:

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

where $\Gamma(\cdot)$ is the Gamma function and V_1 is the volume of the unit d-dimensional hypersphere. The quantity $P(\tilde{r} < r)$ is given by the volume ratio $\frac{V_1}{V_r} = r^d$; moreover, being $P(\tilde{r} < r)$ the cumulative density function (`cdf`) related to the `pdf` $p(r)$, it is $p(r) = \partial \frac{V_r}{V_1} / \partial r = d r^{d-1}$.

The `pdf` $g(r; d, k)$ related to the event $\min_{i \in \{1, \cdots, k\}} \|\mathbf{z}_i\| = r$ (i.e. the minimum distance between the points $\{\mathbf{z}_i\}_{i=1}^k$ and the hypersphere center equals to r) is proportional to the probability of drawing one point with distance r multiplied by that of drawing $k - 1$ points with distance $\tilde{r} > r$,

that is:

$$g(r;d,k) \propto \tilde{g}(r;d,k) = p(r)(1-P(\tilde{r}<r))^{k-1} = \frac{\frac{\partial V_r}{\partial V_1}}{\partial r}\left(1-\frac{V_r}{V_1}\right)^{k-1} = \frac{1}{V_1}dr^{d-1}(1-r^d)^{k-1}$$

Normalizing by $\int_0^1 \tilde{g}(r;d,k)dr = (V_1 k)^{-1}$ we finally get:

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

2.2 Minimum Neighbor Distance Estimators Based on Maximum Likelihood

Having $g(r;d,k)$ we can exploit it to obtain an `i.d.` estimate from a neighborhood of the dataset. Consider a sample set $\mathbf{X}_n = \{\mathbf{x}_i\}_{i=1}^n = \{\psi(\mathbf{z}_i)\}_{i=1}^N \subset \Re^d$, where $\mathbf{z}_i$ are independent identically distributed points drawn from a manifold $\mathcal{M} \equiv \Re^d$ embedded in an higher dimensional space $\Re^D$ through a locally isometric non-linear smooth map $\psi : \mathcal{M} \to \Re^D$; these points are sampled by means of a non-uniform smooth `pdf` $f : \mathcal{M} \to \Re^+$. For each point $\mathbf{x}_i \in \mathbf{X}_N$ we find the set of $k+1 \quad (1 \le k \le N-1)$ nearest neighbors $\hat{\mathbf{X}}_{k+1} = \hat{\mathbf{X}}_{k+1}(\mathbf{x}_i) = \{\mathbf{x}_j\}_{j=1}^{k+1} \subset \mathbf{X}_N$.

Calling $\hat{\mathbf{x}} = \hat{\mathbf{x}}_{k+1}(\mathbf{x}_i) \in \hat{\mathbf{X}}_{k+1}$ the most distant point from $\mathbf{x}_i$, we calculate the distance between $\mathbf{x}_i$ and the nearest neighbor in $\hat{\mathbf{X}}_{k+1}$ and we normalize it by means of the distance between $\mathbf{x}_i$ and $\hat{\mathbf{x}}$. More precisely, we have:

$$\rho(\mathbf{x}_i) = \min_{\mathbf{x}_j \in \hat{\mathbf{X}}_{k+1}} \frac{\|\mathbf{x}_i - \mathbf{x}_j\|}{\|\mathbf{x}_i - \hat{\mathbf{x}}\|}.$$

For $\mathbf{x}_i \neq \hat{\mathbf{x}}$, the quantities $\rho(\mathbf{x}_i)$ are samples drawn from the `pdf` $g(r;d,k) = kdr^{d-1}(1-r^d)^{k-1}$, where the parameter k is known and the parameter d must be estimated. A simple approach for the estimation of d is the maximization of the log-likelihood function:

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

To select an integer value $\hat{d} \in \{1, \cdots, D\}$ as the estimated `i.d.`, it suffices to evaluate

$$\hat{d} = \underset{d \in \{1, \cdots, D\}}{\arg\max}\ ll(d);$$

we call this estimator $\texttt{MiND}_{\texttt{MLi}}$. On the other side, if a real value is required as a fractal `i.d.` estimation, the maximal value in $[1, D]$ must be found. To this aim we compute the first derivative of $ll(d)$ and we determine the solutions of $\frac{\partial ll}{\partial d} = 0$, thus obtaining:

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

We recall that the `MLE` technique adopts a similar derivation since it extracts distance information from all the first k nearest neighbors. We note that, in the particular case of $k = 1$, the solution of the previous equation is:

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

that is exactly the `MLE` estimator proposed in [65] for $k = 1$; we call this estimator $\texttt{MiND}_{\texttt{ML1}}$ and its time complexity is $O(DN \log N)$. For $k > 1$ we numerically solve the following optimization problem:

$$\hat{d} = \underset{1 \leq d \leq D}{\arg\max}\, ll(d).$$

To solve this optimization problem we employed the constrained optimization method proposed in [23] with the initial integer value

$$d_0 = \underset{d \in \{1, \cdots, D\}}{\arg\max}\ ll(d)$$

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

2.2.1 Drawbacks of Local Approaches in High Dimensional Spaces

Unfortunately the number of sample points required to perform the previous dimensionality estimation grows exponentially with the value of the `i.d.`. For this reason, when the dimensionality is too high, the number of sample points practically available is insufficient to compute an acceptable estimation. Moreover, the fraction between the points on (or close to) the

edge of the manifold, and the other points (inside the manifold) increases in probability when the dimensionality increases (the so-called "edge-effect", see [107]), thus affecting the results achieved by estimators based on statistics related to the behaviour of point neighborhoods, such as the algorithms proposed before ($\texttt{MiND}_{\texttt{MLi}}$ and $\texttt{MiND}_{\texttt{MLk}}$) and `MLE`. To formally show this fact, consider a sample $\mathbf{x}_i \in \mathbf{X}_N$ and its k-nearest neighbors, we can prove that $\mathbf{x}_i$ has a very low probability to be located at the center (or to be at least very close to the center) of the hypersphere from which its k-nearest neighbors are supposed to be uniformly drawn. To this aim, we consider $\hat{k} = k + 1$ points uniformly sampled from the centered unit hypersphere; if at least one of these samples is very close to the hypersphere center, that can be referred as $\mathbf{x}_c$, it would be a sample point such that its k-nearest neighbors can be assumed to be uniformly drawn from the unit hypersphere whose center is in (or very close) to $\mathbf{x}_c$ itself. Therefore, we must compute the probability of finding at least one sample, among the $\hat{k}$ we are drawing, which falls at a distance $\tilde{r} < r < 1$ from the hypersphere center $\mathbf{0}_d$. To this purpose, calling $h(\hat{k}; r, d)$ the probability that the $\hat{k}^{th}$ sampled point is the first point drawn at a distance $\tilde{r} < r$ from the hypersphere center, we recall that $h(\hat{k}; r, d)$ is a geometric probability distribution function whose parameter is $p_{geom} = P(\tilde{r} < r)$. According to this, we get the following `pdf`:

$$h(\hat{k}; r, d) = (1 - r^d)^{\hat{k}-1} r^d.$$

A first insight is provided by the consideration that $h(\hat{k}; r, d)$ is an exponential function w.r.t. d, and with base $r < 1$. This means that, having fixed the value of $\hat{k}$ and r, as d grows the probability to get the $\hat{k}^{th}$ sample near the center decreases. Similarly, having fixed the value of k and d, as r becomes smaller the probability to get the $\hat{k}^{th}$ sample at a distance $\hat{r} < r$ from the center decreases. A further consideration is raised by the observation that the expectation of $h(\hat{k}; r, d)$ is $(\frac{1}{r})^d$; this highlights the fact that, on average, the number $\hat{k}$ of neighbors required to finally get the $\hat{k}^{th}$ point at distance $\hat{r} < r$ from the hypersphere center grows exponentially with d. Moreover, we note that the cumulative distribution related to $h(\hat{k}; r, d)$, that is the probability to draw $\hat{k}$ samples such that one of them is a point at distance $\hat{r} < r$ from the hypersphere center, is:

$$H(\hat{k}; r, d) = \sum_{i=0}^{\hat{k}} h(i; r, d) = 1 - (1 - r^d)^{\hat{k}}.$$

Exploiting this equation, and fixing the values of r and $\hat{k}$ (that is $r = 0.1$ and $\hat{k} = 30$), and increasing the `i.d.` value (that is $d = \{2, 5, 10, 50\}$) we see that the value of $H(30; 0.1, d)$ becomes lower and lower:

$$\begin{aligned} H(30; 0.1, 2) &\approx 2.603e^{-02} \\ H(30; 0.1, 5) &\approx 2.999e^{-004} \\ H(30; 0.1, 10) &\approx 3.000e^{-009} \\ H(30; 0.1, 50) &\approx 0 \end{aligned}$$

On the other hand, the value of H increases when the values of $\hat{k}$ and d are fixed, and the value of r is increased. This means that, as the `i.d.` increases, all the sampled point are far from the center, which essentially means that there is no point that could be considered as the center of the hypersphere from which $k = \hat{k} - 1$ nearest neighbors are supposed to be uniformly drawn. To further support our conjecture, we can solve $H(\hat{k}; r, d)$ to compute the number of $\hat{k}$ of nearest neighbors required to sample a point in $\Re^d$ at a distance $\hat{r} < r$ with probability H; more precisely, we obtain:

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

Evaluating this function with fixed values of r and H (that is $r = 0.1, H = 0.9$), and increasing the `i.d.` value ($d = \{2, 5, 10, 30\}$), we obtain increasingly high values of the required number $\hat{k}$ of nearest neighbors:

$$\begin{aligned} \hat{k}(0.1, 0.9, 2) &\approx 229 \\ \hat{k}(0.1, 0.9, 5) &\approx 230257 \\ \hat{k}(0.1, 0.9, 10) &\approx 23025849023 \\ \hat{k}(0.1, 0.9, 50) &\approx \infty \end{aligned}$$

This theoretical and empirical results show that, given a sample point $\mathbf{x}_c$, it can be assumed to be the center of the hypersphere from which its k-nearest neighbors are supposed to be uniformly drawn only when the `i.d.` value is low and the available number of nearest neighbors is high.

2.3 MiND$_{\mathrm{KL}}$: a pdf Comparison Approach

The discussion reported so far shows that neighborhood-based i.d. estimators which assume that the normalized k-nearest neighbors distances resemble the distances between nearest neighbors uniformly sampled from the unit hypersphere, have a well founded theory but lack a proper statistical model. According to the aforementioned results, an i.d. estimator exploiting the normalized k-nearest neighbors distances should adopt a different probability distribution that, to our knowledge, has not been formalized yet.

For these reasons, to obtain a more reliable estimate of the i.d. we propose a novel approach based on the minimization of the Kullback-Leibler divergence between the pdf of the distances of the neighbors points of the dataset and those calculated on synthetic data of known dimensionality.

Notice that, once k is fixed, $g(r;k,d)$ represents a finite family of D pdfs for all the parameters values $1 \leq d \leq D$. Exploiting this fact, another approach for the estimation of the missing parameter d is the comparison between the D possible theoretical pdfs and a density function estimated by means of the given data.

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

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

The function $\hat{g}$ can be obtained by means of a set of sample data points as a parametric model; nevertheless, as stated before, the number of sample points required to perform dimensionality estimation grows exponentially with the value of the i.d.. To reduce the bias between the analytical pdf g and the estimated one $\hat{g}$, for each value $1 \leq d \leq D$ we learn a test pdf $\breve{g}_d(r;k)$ by means of points uniformly draw from the d-dimensional unit hypersphere.

Moreover, to best resemble the point density of the given dataset $\mathbf{X}_N$ of cardinality N, we draw exactly N points per dimensionality. Finally, we numerically estimate the Kullback-Leibler divergence by means of the

estimated $\hat{g}$ and $\breve{g}_d$. More precisely, given a sample set $\mathbf{X}_N = \{\mathbf{x}_i\}_{i=1}^N = \{\psi(\mathbf{z}_i)\}_{i=1}^N \subset \Re^D$ where $\mathbf{z}_i$ are independent identically distributed points drawn from a manifold $\mathcal{M}$ according to a non uniform smooth pdf $f : \mathcal{M} \to \Re^+$, we compute a vector of normalized distances $\hat{\mathbf{r}} = \{\hat{r}_i\}_{i=1}^N = \{\rho(\mathbf{x}_i)\}_{i=1}^N$ using

$$\rho(\mathbf{x}_i) = \min_{\mathbf{x}_j \in \hat{\mathbf{X}}_{k+1}} \frac{\|\mathbf{x}_i - \mathbf{x}_j\|}{\|\mathbf{x}_i - \hat{\mathbf{x}}\|}.$$

For each dimensionality $d \in \{1, \cdots, D\}$ we uniformly drawn a set of N points $\mathbf{Y}_N d = \{\mathbf{y}_i\}_{i=1}^N$ from the unit d-dimensional hypersphere, and we similarly compute a vector of normalized distances

$$\breve{\mathbf{r}}_d = \{\breve{r}_i d\}_{i=1}^N = \{\rho(\mathbf{y}_i)\}_{i=1}^N.$$

Notice that, a d-dimensional vector randomly sampled from a d dimensional hypersphere according to the uniform pdf, can be generated by drawing a point $\bar{\mathbf{y}}$ from a standard normal distribution $\mathcal{N}(\cdot|\mathbf{0}_d, 1)$ and by scaling its norm (see Section 3.29 of [37]):

$$\mathbf{y} = \frac{u^{\frac{1}{d}}}{\|\bar{\mathbf{y}}\|}\bar{\mathbf{y}}, \qquad \bar{\mathbf{y}} \sim \mathcal{N}(\dot{|}\mathbf{0}_d, 1)$$

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

Given a set of values $r_{i=1}^N \subset [0, 1]$ distributed according to the pdf p, in [109] the following estimator is proposed:

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

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

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

where $\hat{\rho}(\hat{r}_i)$ and $\breve{\rho}_d(\hat{r}_i)$ are the distances between $\hat{r}_i$ and its first neighbor in $\hat{\mathbf{r}}$ and in $\hat{\mathbf{r}}_d$ respectively. In [109] a Kullback-Leibler divergence estimator based on the nearest neighbor search is proposed; moreovers, the authors show that their method is more effective than partitioning-based techniques, especially when the number of samples is limited. Employing this estimator

between $\hat{g}$ and $\breve{g}_d$ we obtain:

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

Employing the above equation, the estimated `i.d.` value $\hat{d}$ is computed as follows:

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

We call this estimator $\texttt{MiND}_{\texttt{KL}}$ (`Minimum Neighbor Distance` based on `Kullback-Leibler` divergence); its time complexity is $O(D^2 N \log N)$.

Even if $\texttt{MiND}_{\texttt{KL}}$ obtains reliable results, being able to overcome the limitations of other state-of-the-art methods, it still produces biased estimates when confronted with very high `i.d.` values (i.e. $d > 30$). In order to reduce this shortcoming, in the next paragraph we are going to present the joint use of information related to norms and angles for `i.d.` estimation. Exploiting information related to angles, in addition to the ones related to norms, has proven to provide more precise estimations of the `i.d.` even when this is very high. It is also worth noting that for k neighbors there are $\binom{k}{2}$ pairwise angles that could be used for derive angle-based properties; this permits to reduce the number of required neighbors for having sufficient samples for `i.d.` estimation, thus resulting in neighborhood of smaller size which, as stated before, are more representative of the manifold structure.

2.4 DANCo: Combining Angle and Norm Compressions

So far we described estimators that exploit the information conveyed by the concentration of norms, which still suffer from bias; to overcome this limitation we include the use of the information derived by the concentration of angles. The method that exploits this conjoint information is named `DANCo` (`Dimensionality from Angles and Norms Compression`) and, like $\texttt{MiND}_{\texttt{KL}}$, is based on a `pdf` comparison approach by means of Kullback-Leibler divergences.

2.4.1 A Closed Form of the Distance-Based Kullback-Leibler Divergence

Though the Kullback-Leibler estimation approach proposed by Wang et al. [109] has shown to produce reliable approximation, under our settings the closed-form Kullback-Leibler divergence between two minimum neighbor distance pdfs can be analytically identified. More specifically, once the parameter k is fixed, we firstly need to identify the two pdfs by providing an estimate of the parameter d of $g(\cdot;\cdot,d)$; to accomplish this task, we decided to employ the maximum likelihood estimator $\texttt{MiND}_{\texttt{MLi}}$ described before. Calling $\hat{d}_{ML}$ the ML estimation obtained on the dataset, and $\check{d}_{d,ML}$ the ML estimations computed by means of points sampled from d-dimensional hyperspheres[1] (for $d \in \{1..D\}$), and plugging $\hat{d}_{ML}$ and $\check{d}_{d,ML}$ in g, we obtain two fully defined pdfs whose dissimilarity is measured by computing their Kullback-Leibler divergence. Although there exist distributions which do not admit a closed form of the Kullback-Leibler divergence, its analytical expression for the minimum neighbor distances may be obtained by integration as follows:

$$\overline{KL}_d = \mathcal{KL}(g(\cdot;k,\hat{d}_{ML}), g(\cdot;k,\check{d}_{d,ML})) = \int_0^1 g(r;k,\hat{d}_{ML}) \log\left(\frac{g(r;k,\hat{d}_{ML})}{g(r;k,\check{d}_{d,ML})}\right) \mathrm{d}r$$

$$= \mathcal{H}_k \frac{\check{d}_{d,ML}}{\hat{d}_{ML}} - 1 - \mathcal{H}_{k-1} - \log\frac{\check{d}_{d,ML}}{\hat{d}_{ML}} - (k-1)\sum_{i=0}^{k}(-1)^i \binom{k}{i} \Psi\left(1 + \frac{i\hat{d}_{ML}}{\check{d}_{d,ML}}\right)$$

where $\mathcal{KL}(\cdot,\cdot)$ is the Kullback-Leibler divergence operator, $\mathcal{H}_k$ represents the k-th harmonic number $\left(\mathcal{H}_k = \sum_{i=1}^{k} \frac{1}{i}\right)$, and $\Psi(\cdot)$ is the digamma function.

2.4.2 Angle Compression in High Dimensional Spaces

As it happens for norms, for $\epsilon \to 0^+$, we consider the points of each neighborhood of $\mathcal{M}$ as uniformly drawn from the unit hypersphere. Under these settings, we observe that in high dimensions mutual angles among k uniformly distributed unitary vectors $\{\mathbf{x}_i\}_{i=1}^{k}$ on a $(d-1)$-dimensional surface S^{d-1} of a hypersphere in $\Re^d$ are subject to the concentration of their values.

[1] Note that, even if the ML estimates $\check{d}_{d,ML}$ are biased w.r.t. the real value d employed in the sampling process, due to the kNN bias effect described above, a comparable bias can also be observed in the estimated $\hat{d}_{ML}$; this is the reason why the Kullback-Leibler estimation approach is not affected by this distortion.

The common belief that in high dimensions such vectors tend to be orthogonal to each other has found partly justification in the past [74], but only in the last decades an even deeper investigation has allowed a more precise characterization of this fact [92].

Dealing with angles subtended by bidimensional vectors in a circle, or more generally with directions of unit vectors in $\Re^d$, opens the way to the field of circular and directional statistics. In particular, two of the most adopted distributions therein are the von Mises distribution (`VM`) and its high-dimensional generalization termed von Mises-Fisher distribution (`VMF`, [74]). More precisely, for $\mathbf{x} \in S^{d-1}$, the `VMF` distribution with parameters ν and τ has the following density function:

$$q(\mathbf{x}; \nu, \tau) = C_d(\tau) \exp\left(\tau \nu^T \mathbf{x}\right)$$

where the unit vector ν denotes the mean direction, and the concentration parameter $\tau \geq 0$ gets high values in case of a high concentration of the distribution around the mean direction. In particular, $\tau = 0$ when points are uniformly distributed on S^{d-1}. Moreover, the normalization constant $C_d(\tau)$ in the above equation takes the following form:

$$C_d(\tau) = \frac{\tau^{d/2-1}}{(2\pi)^{d/2} I_{d/2-1}(\tau)}$$

where I_v is the modified Bessel function of the first kind with order v. Due to the normalization factor, this `pdf` is difficult to be used in theoretical derivations; moreover, following our assumptions, no information about d may be estimated by the knowledge of the parameters ν and τ, being ν uninformative when the hyper-solid angles are uniformly distributed ($\tau = 0$), as in uniformly sampled hyperspheres.

Therefore, to infer the `i.d.` of $\mathcal{M}$ by exploiting angular information, we focus on the distribution of the angles θ computed between independent pairs of random points chosen in the neighborhoods of $\Re^d$ and sampled from the uniform distribution in the hypersphere (which will be referred to as pairwise angles in the following). Note that working on pairwise angles allows both to exploit the concentration factor τ, which is strictly related to the dimensionality d as we will show, and to rely on the `VM` distribution, which is more tractable compared to the `VMF pdf`.

Considering the angle $\theta \in [-\pi, \pi]$ between two vectors, the `VM pdf` of θ

reads as:

$$q(\theta;\nu,\tau) = \frac{\mathrm{e}^{\tau\cos(\theta-\nu)}}{2\pi I_0(\tau)}\chi_{[-\pi,\pi]}(\theta)$$

with the same parameters and notation adopted for the VMF pdf. Intuitively, the VM distribution is the circular counterpart of the normal distribution on a line, sharing with the latter many interesting properties [9]. To understand the link between τ and d, we recall that $q(\theta;\nu,\tau)$ is unimodal for $\tau > 0$, as a Gaussian random variable peaked around its mean. Furthermore, we introduce a theorem to show that increasing values of τ are expected for points uniformly drawn from hyperspheres with increasing dimensionality d.

Theorem 2.2. *Given two independent random unit vectors* $(\mathbf{x}_1, \mathbf{x}_2)$ *in* $\Re^d$*, drawn from a uniform distribution on* S^{d-1}*, for increasing values of the dimensionality* d*, the concentration parameter* τ *of the VM distribution describing the angle* θ *between* $\mathbf{x}_1$ *and* $\mathbf{x}_2$ *converges asymptotically to the dimensionality* d*.*

Proof. Consider the following results:

1. for $d \to \infty$, the random variable $\tilde{\theta} = \sqrt{d}(\theta - \frac{\pi}{2})$ converges in distribution to a standard normal pdf (see Lemma 3.1 in [92]). In other words, θ converges in distribution to a Gaussian random variable with mean $\frac{\pi}{2}$ and standard deviation $\frac{1}{\sqrt{d}}$;

2. for large concentration values τ, a VM distribution with parameters ν and τ can be approximated by a Gaussian distribution with mean ν and standard deviation $\frac{1}{\sqrt{\tau}}$ [102]. Several approximations have been proposed since the mid-century, all of which are based on the observation that, thanks to the asymptotic forms of the Bessel function [1], for values of $\tau > 10$ the distribution of the random variable $\theta\sqrt{\tau}$ may be approximated by a standard normal distribution [74]. More accurate approximations [49] sharing the same asymptotic behavior have been introduced by considering further terms in the power series expansion of $\cos(\theta - \nu)$ in $q(\theta;\nu,\tau)$.

Item 1 guarantees that θ converges in distribution to a Gaussian random variable with mean $\frac{\pi}{2}$ and standard deviation $\frac{1}{\sqrt{d}}$. Since $\lim_{d\to+\infty}\frac{1}{\sqrt{d}} = 0$, we can assume that, for sufficiently high values of d, the angles θ concentrate

on their mean. Therefore, we could describe the distribution of θ by means of a VM distribution whose concentration parameter τ takes large values. At this point, we can apply item 2 ensuring that θ converges in distribution to a Gaussian pdf with mean $\frac{\pi}{2}$ and standard deviation $\frac{1}{\sqrt{\tau}}$. It follows that $\tau \asymp d$, namely $\lim_{d\to+\infty} \frac{\tau}{d} = 1$. □

This theorem has both a general and a specific value. At first, it formally proves the existence of the concentration of angles in high dimensions, stating both an asymptotic linear relation between concentration and dimensionality, and the orthogonality between any couple of infinite-dimensional vectors. Secondly, it allows to estimate the i.d. of the observed points through the estimation of the concentration parameter τ.

As a further advantage, it suggests that, having to cope with high dimensional manifolds, the i.d. estimate computed by exploiting the concentration of angles is reliable and can be used to enforce the i.d. estimate obtained by employing the concentration of norms. Unfortunately, the same finiteness of the sample size which prevents nearest distance-based estimators from performing well in practical scenarios, limits the aforementioned advantage, even though here we may rely on a more considerable number $\binom{k}{2}$ of pairwise angles within each kNN upon which to base our estimate. This is why the novel estimator we are going to propose, called DANCo employs both the ML estimation of the VM parameters ν and τ, and the Kullback-Leibler divergence between the VM pdf estimated from the observed dataset and those computed on synthetic data of known i.d.s.

2.4.3 A Closed Form of the Angle-Based Kullback-Leibler Divergence

Assuming that $\{\theta_1, \ldots, \theta_N\}$ is a sample drawn from a VM distribution with parameters ν and τ, the ML of ν equals the sample mean direction, that is:

$$\hat{\nu} = \operatorname{atan}_2 \left(\sum_{i=1}^{N} \sin\theta_i, \sum_{i=1}^{N} \cos\theta_i \right)$$

where $\operatorname{atan}_2(x, y)$ is the standard operator computing the arc tangent of y/x, taking into account which quadrant the point (x, y) lies in. This kind of non euclidean mean operator is commonly used when circular quantities are involved in the computation.

Likewise, the ML of τ equals the concentration parameter $\hat{\tau}$ calculated as a solution of $\eta = \frac{I_1(\tau)}{I_0(\tau)} \equiv A(\tau)$, being η the norm of the sample mean vector defined by Upton [103] as:

$$\eta = \sqrt{\left(\frac{1}{N}\sum_{i=1}^{N}\cos\theta_i\right)^2 + \left(\frac{1}{N}\sum_{i=1}^{N}\sin\theta_i\right)^2}$$

Being A a non invertible function, we rely on the well-known and qualified method proposed in [36], which approximates $A^{-1}(\eta)$ by:

$$\hat{\tau} = \widetilde{A}^{-1}(\eta) = \begin{cases} 2\eta + \eta^3 + \frac{5\eta^5}{6} & \eta < 0.53 \\ -0.4 + 1.39\eta + \frac{0.43}{1-\eta} & 0.53 \leq \eta < 0.85 \\ \frac{1}{\eta^3 - 4\eta^2 + 3\eta} & \eta \geq 0.85 \end{cases}$$

Once an estimate of the VM pdf is obtained, we need to compare it with those computed on synthetic data of known i.d.s. To this aim, a closed-form of the Kullback-Leibler divergence between two VM pdfs of parameters ν_1, τ_1, and ν_2, τ_2 is defined in [108] as:

$$\begin{aligned} \overline{KL}_{\nu,\tau} &= \mathcal{KL}(q(\cdot;\nu_1,\tau_1), q(\cdot;\nu_2,\tau_2)) = \int_{-\pi}^{\pi} q(\theta;\nu_1,\tau_1)\log\left(\frac{q(\theta;\nu_1,\tau_1)}{q(\theta;\nu_2,\tau_2)}\right)\mathrm{d}\theta \\ &= \log\frac{I_0(\tau_2)}{I_0(\tau_1)} + \frac{I_1(\tau_1) - I_1(-\tau_1)}{2I_0(\tau_1)}\left(\tau_1 - \tau_2\cos(\nu_2 - \nu_1)\right) \end{aligned}$$

Note that the introduced framework can be applied for i.d. estimation only if the pdf of angles θ in the embedding space $\Re^D$ converges to the pdf q related to the VM distribution. This is guaranteed by the local isometry of the map $\phi : \Re^d \to \Re^D$ embedding a dataset drawn from a manifold $\mathcal{M} = \Re^d$ in a higher dimensional space $\Re^D$. In fact, the local isometry of ϕ guarantees its conformality with constant dilation factor equal to 1 [78], which intuitively means that a distance preserving map has also the property of preserving angles, as long as the overall area is maintained.

2.4.4 A pdf Comparison Approach Expoliting Norms and Angles

Unfortunately, even informations related on angles suffer from a severe bias strictly connected with the employment of the kNN method (on a finite set of points) which, in turn, violate almost in part the assumptions introduced in the previous sections. This behavior is intuitively depicted in figure 2.3,

where we observe a systematic positive bias in the angle-based `i.d.` estimation which is only loosely counterbalanced by a regular `i.d.` underestimation based on norms (see figure 2.3 (a)). We may read this duality in terms of an opposite behaviour of the sensitivity of the angles' compression w.r.t. the dataset `i.d.` when compared to the norm one, especially in high dimensions (see figure 2.3 (b)).

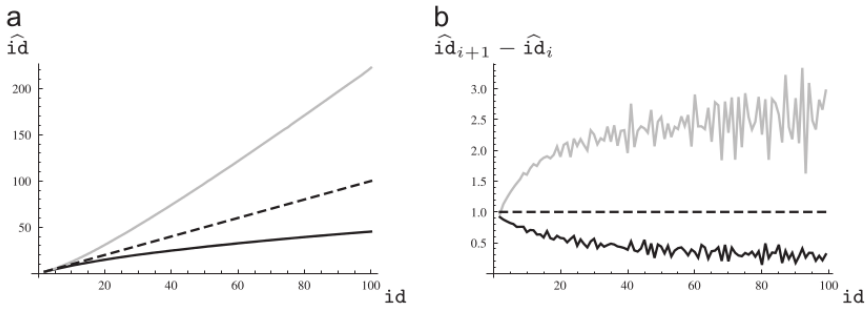

Figure 2.3: Comparison between: (a) the `i.d.` estimates based on angles (gray curve) and norms (black curve) with the exact `i.d.` values (dashed curve), and (b) the finite differences of the `i.d.` estimates.

In search of an unbiased `i.d.` estimator, a profitable joint use of the information derived by both angles and norms demands special attention and requires techniques which goes beyond suitable aggregations of the two estimates. Namely, a successful strategy consists in comparing the joint `pdf` $\hat{h}(r,\theta)$ of the nearest neighbor distances r and pairwise angles θ related to the real dataset with the D `pdf`s computed on samples drawn from hyperspheres of increasing dimensionality, which will be referred to as $h_d(r,\theta)$ in the following (for $d \in \{1..D\}$). Summarizing, the `i.d.` estimate we want to compute is:

$$\hat{d} = \underset{d\in\{1..D\}}{\arg\min} \int_{-\pi}^{\pi}\int_{0}^{1} h_d(r,\theta)\log\left(\frac{h_d(r,\theta)}{\hat{h}(r,\theta)}\right)\mathrm{d}r\mathrm{d}\theta$$

Since the norm distribution $g(r;k,d)$ and the angle distribution $q(\theta;\nu,\tau)$ are independent when the data are uniformly drawn from a spherical distribution [72], the joint `pdf` factorizes in the product of the two marginals, i.e. $h_d(r,\theta) = g(r;k,d)q(\theta;\nu,\tau)$, so that the Kullback-Leibler divergence $\overline{KL}_{d,\nu,\tau}$ between $h_d(r,\theta)$ and $\hat{h}(r,\theta)$ may be split in the sum of the two closed-form

divergences reported before, as follows:

$$\overline{KL}_{d,\nu,\tau} = \mathcal{KL}(h_d(r,\theta), \hat{h}(r,\theta)) = \overline{KL}_d + \overline{KL}_{\nu,\tau}$$

Therefore, an `i.d.` estimator based on the the above equation is obtained by finding the dimensionality d minimizing the Kullback-Leibler divergence $\overline{KL}_{d,\nu,\tau}$, hence:

$$\hat{d} = \underset{d\in\{1..D\}}{\arg\min}\, \overline{KL}_{d,\nu,\tau}$$

2.4.5 DANCo

Under the same theoretical setting as before, we realized an `i.d.` estimator, called `DANCo`, which exploits the information conveyed by the compression of norms and angles. We firstly extract the information conveyed by the concentration of norms by working on the neighborhood of each point in the dataset. More specifically, for each $\mathbf{x}_i \in \mathbf{X}_N$, we extract the set of its $k+1$ ($1 \leq k \leq N-2$) nearest neighbors $\bar{\mathbf{X}}_{k+1} = \bar{\mathbf{X}}_{k+1}(\mathbf{x}_i) = \{\mathbf{x}_j\}_{j=1}^{k+1} \subset \mathbf{X}_N$. Calling $\hat{\mathbf{x}} = \hat{\mathbf{x}}_{k+1}(\mathbf{x}_i) \in \bar{\mathbf{X}}_{k+1}$ the farthest neighbor of $\mathbf{x}_i$, we calculate the distance between $\mathbf{x}_i$ and its nearest neighbor in $\bar{\mathbf{X}}_{k+1}$, and we normalize it by means of the distance between $\mathbf{x}_i$ and $\hat{\mathbf{x}}$, that is:

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

This equation is used to compute the vector $\hat{\mathbf{r}} = \{\hat{r}_i\}_{i=1}^N = \{\rho(\mathbf{x}_i)\}_{i=1}^N$ of normalized distances. After this, we compute the `ML` estimation by numerically solving the optimization problem as seen before,

$$\hat{d}_{ML} = \underset{1\leq d\leq D}{\arg\max}\, ll(d)$$

where:

$$ll(d) = N\log kd + (d-1)\sum_{\mathbf{x}_i\in\mathbf{X}_N} \log\rho(\mathbf{x}_i) + (k-1)\sum_{\mathbf{x}_i\in\mathbf{X}_N} \log\left(1-\rho^d(\mathbf{x}_i)\right)$$

To this aim we employ the constrained optimization method proposed in [23] with the initial (integer) value $d_0 = \arg\max_{d\in\{1..D\}} ll(d)$.

Similarly, we analyze local neighborhoods of the dataset to capture the information provided by the concentration of pairwise angles; in particular, for each point $\mathbf{x}_i \in \mathbf{X}_N$ we find its k nearest neighbors $\bar{\mathbf{X}}_k^i$ and we center them by means of a translation to obtain $\hat{\mathbf{X}}_k^i = \left\{\mathbf{x}_j - \mathbf{x}_i : \forall\, \mathbf{x}_j \in \bar{\mathbf{X}}_k^i\right\}$. Next, we employ the following function:

$$\theta(\mathbf{x}_z, \mathbf{x}_j) = \arccos \frac{\mathbf{x}_z \cdot \mathbf{x}_j}{\|\mathbf{x}_z\|\|\mathbf{x}_j\|}$$

to calculate the $\binom{k}{2}$ angles of all the possible pairs of vectors in $\hat{\mathbf{X}}_k^i$; in this way, for each neighborhood, we compute a vector $\hat{\theta}_i = \{\theta(\mathbf{x}_z, \mathbf{x}_j) : \forall\, \mathbf{x}_z,\ \mathbf{x}_j \in \hat{\mathbf{X}}_k^i\}_{1 \leq z < j \leq k}$.

Since each component of $\hat{\theta}_i$ follows a `VM pdf` of parameters ν and τ, for each set of neighbors we estimate their values by employing the `ML` approach described before, thus obtaining the vectors $\hat{\nu} = \{\hat{\nu}_i\}_{i=1}^N$ and $\hat{\tau} = \{\hat{\tau}_i\}_{i=1}^N$. Finally, we compute their means $\hat{\mu}_\nu = \text{atan}_2(\sum_{i=1}^N \sin\hat{\nu}_i, \sum_{i=1}^N \cos\hat{\nu}_i)$ and $\hat{\mu}_\tau = N^{-1}\sum_{i=1}^N \hat{\tau}_i$.

At this point, the statistics extracted from the input dataset must be compared with those computed on synthetic datasets of known `i.d.`. Therefore, for each dimensionality $d \in \{1..D\}$ we uniformly draw a set of N points $\mathbf{Y}_{Nd} = \{\mathbf{y}_i\}_{i=1}^N$ from the unit d-dimensional hypersphere (named $\texttt{hs}^d$-sample in the following), and we employ them to compute the vector of normalized distances $\check{\mathbf{r}}_d = \{\check{r}_{id}\}_{i=1}^N = \{\rho(\mathbf{y}_i)\}_{i=1}^N$ and its `ML` estimation $\check{d}_{d,ML}$. Next, we calculate the vectors of the `VM` distribution parameters $\check{\nu}_d = \{\nu_i\}_{i=1}^N$ and $\check{\tau}_d = \{\tau_i\}_{i=1}^N$ together with their means $\check{\mu}_\nu^d$ and $\check{\mu}_\tau^d$.

Finally, we compose the Kullback-Leibler divergences, thus obtaining the following `i.d.` estimate:

$$\hat{d} = \underset{d \in \{1..D\}}{\arg\min}\, \mathcal{KL}(g(\cdot; k, \hat{d}_{ML}), g(\cdot; k, \check{d}_{d,ML})) + \mathcal{KL}(q(\cdot; \hat{\mu}_\nu, \hat{\mu}_\tau), q(\cdot; \check{\mu}_\nu^d, \check{\mu}_\tau^d))$$

For the sake of clarity, we report its pseudo-code in the Appendix A. As shown therein, a further conditional statement has been introduced in order to check whether the `i.d.` estimate computed through the sole normalized nearest neighbor distances falls below 2. In such case, as no pairwise angles can be computed in domains of dimension less than 2, we solely rely on the `i.d.` estimate provided by the aforementioned distances, which can be profitably used in estimating low `i.d.`s without suffering from the usual drawbacks affecting high dimensions.

The time complexity of DANCo is $O(D^2 N \log N)$ and it is dominated by the time complexity of the kNN algorithm ($O(DN \log N)$). The square dependency on the embedding space dimension D is due to the computation of the kNN on each $\mathtt{hs}^d$-sample of growing dimensionality $d \in \{1..D\}$.

2.4.6 A Fast Implementation of DANCo

Although outperforming state-of-the-art algorithms, as shown in [18], a drawback of DANCo is the long time spent for computing the kNN graph on the $\mathtt{hs}^d$-samples for $d = \{1..D\}$, especially for large values of the embedding space dimension D. As the overall procedure relies on the kNN graph, the only way to reduce it (apart from either speeding up the kNN computation through fast algorithms or relying on parallel implementations which exploit, for instance, the high flexibility of modern GPUs) is to avoid, or at least limit, the computation of the kNN on all the $\mathtt{hs}^d$-samples.

To this end, we firstly note that the connection between the dataset in question and the $\mathtt{hs}^d$-samples is extremely loose. In fact, being each $\mathtt{hs}^d$-sample uniformly drawn from the unit d-dimensional hypersphere, to generate it we only need to know the neighboring size k, the sample size N, and the dimensionality d of the hypersphere. As DANCo proves to be robust against changes in k, after having fixed its value we can generate $\mathtt{hs}^d$-samples for different dimensionalities d and sample sizes N, precomputing the associated statistics $[\check{d}_{d,ML}, \check{\mu}_\nu^d, \check{\mu}_\tau^d]$ according to the procedure depicted in the previous section. As shown in figure 2.4, the regularity of the trend of $[\check{d}_{d,ML}, \check{\mu}_\nu^d, \check{\mu}_\tau^d]$ w.r.t. d and N may be fruitfully described through suitable fitting functions.

This not only has the obvious advantage of reducing the time spent in computing the aforementioned statistics, but has also the merit of avoiding those estimate oscillations we observe in DANCo which, though rare, would pose a threat to its performances. In fact, the smoothness of the fitting surface is further enforced by the generation of a given number of $\mathtt{hs}^d$-samples for each dimension d and sample size N (in our case we used 35 replicas), and by the subsequent averaging to obtain more regular $\mathtt{hs}^d$-samples.

Regarding the selection of the fitting function, we are free to choose any function general enough to approximate the data, and sufficiently smooth to avoid overfitting; the data regularity and several tests on robustness, generalization, and accuracy of the extrapolation, lead us to work with cubic

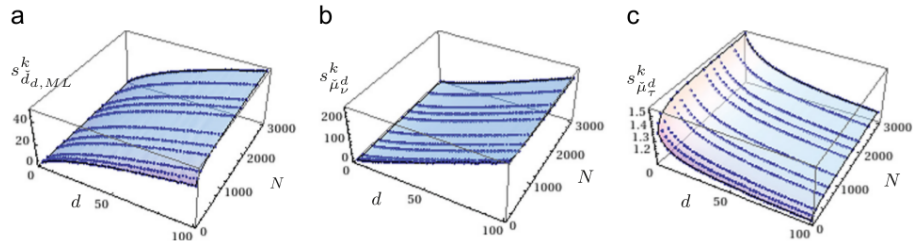

Figure 2.4: The cubic smoothing splines fitting the points: (a) $\check{d}_{d,ML}$, (b) $\check{\mu}^d_\nu$, and (c) $\check{\mu}^d_\tau$ w.r.t. the dimensionality d and the sample size N for $k = 10$, and averaging over 35 replicas of each $\texttt{hs}^d$-sample.

smoothing splines. Figure 2.4 shows the splines $s^k_{\check{d}_{d,ML}}$, $s^k_{\check{\mu}^d_\nu}$, and $s^k_{\check{\mu}^d_\tau}$ fitting, respectively, $\check{d}_{d,ML}$, $\check{\mu}^d_\nu$, and $\check{\mu}^d_\tau$ for different dimensionalities d and sample sizes N.

The introduction of the fitting functions allows to bypass all these steps used in `DANCo` to both generate the $\texttt{hs}^d$-samples and compute the related statistics. As a result, excluding from the time analysis the precomputation of the fitting functions that may be performed once for all (once k is fixed) in an off-line mode, the time complexity of this fast technique, called `FastDANCo`, is $O(DN \log N)$ (its pseudo-code is reported in Appendix A). Note that the use of the fitting functions does not affect the system effectiveness since the results achieved by `FastDANCo` are comparable with those of `DANCo`, and in some cases they prove to be even better.

Fast-kNN

It is possible to further reduce the time complexity of the proposed algorithms optimizing the method employed to build the `kNN` graph, since the `kNN` graph construction technique by brute-force has time complexity $O(DN^2)$, thus representing the most computationally expensive part of our methods. To this aim, some interesting approaches have been proposed in literature, including two methods proposed by Paredes et al. [80], where the authors presented a `kNN` graph construction for general metric spaces, whose empirical time complexity is low. Unfortunately, both the proposed methods require a global data structure and are therefore difficult to be parallelized across machines. Other two efficient methods for the Euclidean

metric space, have been recently developed, which are based on space filling curves [24] and recursive data partitioning [19].

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

These algorithms are structured as follows: the divide step uses an inexpensive Lanczos procedure to perform recursive spectral bisection, and then, after each conquer step, an additional refinement is performed to improve the accuracy of the graph. Note that a hash table is continuously updated to avoid repeating distance calculations during the divide and conquer process.

The strong difference between the two algorithms is in the divide step; indeed, while $\texttt{kNN}_{glue}$ splits the set into three subsets, $\texttt{kNN}_{overlap}$ divides the set into two subsets. Both of these novel kNN constructions don't affect the quality of our estimators, markedly reducing the time needed to execute them.

Chapter 3

Comparison and Benchmark

As shown in Chapter 1, `i.d.` estimators proposed in literature have been evaluated on different datasets, making the comparison of their performance almost impossible. This highlights the need of a benchmark framework to objectively assess and compare different techniques in terms of robustness, w.r.t. parameter settings, large dimensional datasets, noisy datasets, and so on. In this chapter, after recalling experimental datasets and evaluation procedures introduced in literature, we choose some of them to propose a benchmark framework 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 `i.d.` estimators whose code is publicly available, with the techniques that we described so far.

This chapter is organized as follows: in section §3.1 and section §3.2 datasets and methods used to evaluate the performance of the `i.d.` estimators are described. In section §3.3 the standard framework is proposed, while in section §3.4 the benchmark framework is exploited to evaluate some of the cited methods which are either well-known or recent and promising. A statistical analysis of the experimental results is also given.

3.1 Datasets

In the literature both synthetically generated datasets and real dataset are used to evaluate the `i.d.` estimate of the proposed methods. Synthetic datasets are generated by drawing samples from manifolds ($\mathcal{M}$) of known dimension linearly or non-linearly embedded in higher dimensional spaces. It's worth noting that for some real datasets the value of `i.d.` is not known

but only evaluated in a specific range, based on a consensus of previous results of i.d. estimators. In the following we describe those we choose to use in our benchmark study.

Synthetic Datasets

The publicly available tool[1] proposed by Hein in [45] allows to generate 13 kinds of synthetic datasets by uniformly drawing samples from 13 manifolds of known i.d.; they are schematically described in table 3.1, where they are referred to as $\mathcal{M}_*^H$. These manifolds are embedded in higher dimensional spaces through both linear and non-linear maps and are characterized by different curvatures. We note that manifold $\mathcal{M}_8^H$ is particularly challenging for its high curvature; indeed, when it is used for testing, most relevant i.d. estimators compute pronounced i.d. overestimates (see also the results reported in [88]).

Another interesting dataset [11] is generated by sampling a d-dimensional paraboloid, $\mathcal{M}_{Pd}$, 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 $\mathbf{M}_{beta}$, 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 $\mathbf{X}_N \in \Re^{N\times 10}$; multiply each point of $\mathbf{X}_N$ ($\mathbf{X}_N(i,j)$) by $\sin(\cos(2\pi\mathbf{X}_N(i,j)))$, thus obtaining a matrix $\mathbf{D}_1 \in \Re^{N\times 10}$; multiply each point of $\mathbf{X}_N$ by $\cos(\sin(2\pi\mathbf{X}_N(i,j)))$, thus obtaining another matrix $\mathbf{D}_2 \in \Re^{N\times 10}$; append $\mathbf{D}_1$ and $\mathbf{D}_2$ to generate a matrix $\mathbf{D}_3 \in \Re^{2500\times 20}$; append $\mathbf{D}_3$ 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 i.d., we propose to generate two datasets, referred to as $\mathbf{M}_{N1}$ and $\mathbf{M}_{N2}$ in the following. To generate $\mathbf{M}_{N1}$ we uniformly draw N points in $[0,1]^{18}$, we transform each point by means of $\tan(\mathbf{x}^i\cos(\mathbf{x}^{18-i+1}))$ where $i = 1,\cdots,18$, we obtain points in $\Re^{36}$ by appending each transformed $\mathbf{x}$ to $\arctan(\mathbf{x}^{18-i+1}\sin(\mathbf{x}^i))$, we duplicate the coordinates of each point to finally generate points in $\Re^{72}$. The i.d. of $\mathbf{M}_{N1}$ is 18, and its points are drawn

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

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

Table 3.1: The 13 types of synthetic datasets generated with the tool proposed in [45].

from a manifold nonlinearly embedded in $\Re^{72}$. To generate $\mathbf{M}_{N2}$ containing N points in $\Re^{96}$, we applied the same procedure on vectors sampled in $[0,1]^{24}$.

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 [98], `MNIST` database [63], `Isolet` dataset [38], $D2$ `Santa Fe` [83] dataset, and `DSVC1` time series [13]. Recently, the Crystal Fingerprint space for the chemical compound silicon dioxide dataset has also been proposed [104].

`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 3.1, first row).

`MNIST` database consists in 70000 gray-level images of size 28×28 of handwritten digits (see figure 3.1, second row). The real `i.d.` of this database is not actually known, but some works [45, 28] propose similar estimates for the different digits; as an example, the proposed `i.d.` 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 $isolet1$, $isolet2$, $isolet3$, $isolet4$, and $isolet5$. The real `i.d.` value characterizing this dataset is not actually known, but a study reported in [60] 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 (`i.d.` $= 9$) 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 [79], 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 [22]. Employing the method of delays with $D = 20$, a dataset containing 250 points in $\Re^{20}$ is obtained. The `i.d.` characterizing this dataset is ~ 2.26 [13].

Crystal Fingerprint spaces, or Crystal Fingerspaces, have been recently proposed in crystallography [104] with the aim of representing crystalline structures; these spaces are built starting from the measured distances between atoms in the crystalline structure. The theoretical `i.d.` 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 3.1: (First row) Samples from `ISOMAP` face database. (Second row) Samples from digit '0' to digit '9' in `MNIST` database.

3.2 Estimator Evaluation Methods

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

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 [45, 27, 26, 28], thus obtaining a distinct performance evaluation measure for each cardinality.

The second approach estimates the `i.d.` on T permutations of the same dataset and averages the T `i.d.` estimates to obtain the final one [71, 88, 65, 18]. This value is then compared with the real one to assess the `i.d.` estimator.

To also test the estimator's robustness w.r.t. its parameter settings, in [65, 71, 88] the authors apply a further test, originally proposed by Levina et al. in [65]. Particularly, 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 `i.d.` estimate as a function of both the dataset's cardinality and the parameter settings.

Note that, since `i.d.` estimators are usually tested on different datasets to evaluate their reliability when confronted by different dataset structures and configurations, in [71] an overall evaluation measure is proposed. This indicator, called Mean Percentage Error (`MPE`), summarizes all the obtained results in a unique value computed as: $\texttt{MPE} = \frac{100}{\#\mathbf{M}} \sum_{\mathbf{M}} \frac{|\hat{d}_{\mathbf{M}} - d_{\mathbf{M}}|}{d_{\mathbf{M}}}$, where $\#\mathbf{M}$ is the number of tested datasets, $\hat{d}_{\mathbf{M}}$ is the `i.d.` estimated on the dataset

$\mathbf{M}$, and $d_{\mathbf{M}}$ is the real `i.d.` of $\mathbf{M}$. To apply this technique to real datasets whose `i.d.` belongs to the range $\{d_{min}..d_{max}\}$, the associated `MPE`'s term is calculated as: $\min_{d \in \{d_{min}..d_{max}\}} \left(\frac{|\hat{d}_{\mathbf{M}} - d|}{d_{\mathbf{M}}} \right)$, where $d_{\mathbf{M}}$ is the mean of the range.

Finally, to test the significance of differences in performance of the tested methods, we rely on the safe and robust non-parametric Friedman test (`FT`) followed by a wide family of post-hoc tests to effectively check if and which technique overperform the examined competitor algorithms.

3.3 A New Standard Framework

In this section we propose an evaluation approach which can be used as a standard framework to assess estimators performance, comparing it to relevant `i.d.` estimators whose code is publicly available. In this benchmark, we suggest to use the following estimators: `Hein`, `MLE`, `kNNG`, `MLSVD`, `BPCA`, `CD`, $\texttt{MiND}_{\texttt{KL}}$, and `DANCo`[2].

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 `i.d.` is a user-defined parameter should be created with sufficiently high `i.d.` values (`i.d.` ≥ 12).

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 [52].

   c) perform the tests proposed in [65] to evaluate the robustness, w.r.t different cardinalities and parameter settings.

[2]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,
$\texttt{MiND}_{\texttt{KL}}$, and `DANCo`: http://www.mathworks.it/matlabcentral/fileexchange/40112-intrinsic-dimensionality-estimation-techniques

The 21 synthetic datasets used in the benchmark, referred to as $\mathbf{M}_*$ in the following, are listed in table 3.2 with their relevant characteristics (N, d, and D). The first 15 datasets are generated with the tool proposed in [45]; they include 4 instances, $\mathbf{M}_{10*}$, of dataset $\mathbf{M}_{10}$, which are drawn from $\mathcal{M}_{10}^H$ after its embedding in $\Re^D$ by setting $D = \{11, 18, 25, 71\}$. Note that we did not include the dataset sampled from $\mathcal{M}_8^H$ (see table 3.1) since relevant and recent `i.d.` estimators have similarly produced highly overestimated results when tested on it [88]. Indeed, dealing with highly curved manifolds is still a quite challenging problem in the field.

The last six synthetic datasets are $\mathbf{M}_{N1}$, $\mathbf{M}_{N2}$, $\mathbf{M}_{beta}$, and 3 instances of dataset $\mathbf{M}_{P*}$, which are sampled from paraboloids $\mathcal{M}_{Pd}$ whose `i.d.` 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.

Regarding the real datasets we used the `DSVC1` time series [13] ($\mathbf{M}_{\texttt{DSVC1}}$, `i.d.` ~ 2.26), the `ISOMAP` face database [98] ($\mathbf{M}_{\texttt{ISOMAP}}$, `i.d.` $= 3$), the `Santa Fe` dataset [83] ($\mathbf{M}_{\texttt{SantaFe}}$, `i.d.` $= 9$), the `MNIST` database [63] ($\mathbf{M}_{\texttt{MNIST1}}$, `i.d.` $\in \{8..13\}$), the `Isolet` dataset [38] ($\mathbf{M}_{\texttt{Isolet}}$, `i.d.` $\in \{16..22\}$), and the Crystal Fingerprint space for the chemical compound silicon dioxide SiO_2 structure with 3 atoms (this allows to obtain the $\mathbf{M}_{\texttt{SiO2}}$ dataset containing 4738 points embedded in $\Re^{1800}$, and being characterized by an `i.d.` equal to 12).

To run multiple tests also on $\mathbf{M}_{\texttt{MNIST1}}$, $\mathbf{M}_{SiO_2}$, and $\mathbf{M}_{\texttt{Isolet}}$, for each of them we generated 5 instances by extracting random subsets containing 2500 points each and we averaged the achieved results.

The parameter values we employed for different estimators are summarized in table 3.3. Note that, to relax the dependency of the `kNNG` algorithm from the setting 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 $\texttt{kNNG}_1$ and $\texttt{kNNG}_2$) obtained by varying the parameters M and N.

3.4 Experimental Results

results

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

Table 3.2: Synthetic datasets and real datasets suggested by the benchmark; **N** is the dataset cardinality, **d** is the `i.d.`, and **D** is the embedding space dimension.

The results obtained by the compared estimators on the synthetic datasets are summarized in table 3.4 summarizes, while in table 3.5 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), $\texttt{MiND}_{\texttt{KL}}$ (6 out of 21), `BPCA` (4 out of 21), and `MLE` (1 out of 21). It can be further noted that $\texttt{kNNG}_*$, `CD`, `MLE`, and `Hein` obtain good estimates only for low `i.d.` manifolds, while they

Dataset	Method	Parameters
Synthetic	MLE	$k_1 = 6\ k_2 = 20$
	DANCo	$k = 10$
	$\texttt{kNNG}_1$	$k_1 = 6, k_2 = 20, \gamma = 1, M = 1, N = 10$
	$\texttt{kNNG}_2$	$k_1 = 6, k_2 = 20, \gamma = 1, M = 10, N = 1$
	BPCA	$iters = 2000,\ \alpha = (2.0, 2.0)\ \pi = (2.0, 2.0)\ \mu = (0.0, 0.01)$
	Hein	$None$
	CD	$None$
	MLSVD	$None$
	$\texttt{MiND}_{\texttt{KL}}$	$k = 10$
Real	MLE	$k_1 = 3\ k_2 = 8$
	DANCo	$k = 5$
	$\texttt{kNNG}_1$	$k_1 = 3, k_2 = 8, \gamma = 1, M = 1, N = 10$
	$\texttt{kNNG}_2$	$k_1 = 3, k_2 = 8, \gamma = 1, M = 10, N = 1$
	BPCA	$iters = 2000,\ \alpha = (2.0, 2.0)\ \pi = (2.0, 2.0)\ \mu = (0.0, 0.01)$
	Hein	$None$
	CD	$None$
	MLSVD	$None$
	$\texttt{MiND}_{\texttt{KL}}$	$k = 5$

Table 3.3: 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.

produce underestimated values when processing datasets of high i.d..

By observing the MPE indicator, which accounts for the precision of the achieved estimates, we obtain a different ranking: DANCo, $\texttt{MiND}_{\texttt{KL}}$, $\texttt{kNNG}_1$ and $\texttt{kNNG}_2$, MLE, Hein, CD, and MLSVD. This difference is due to the fact that algorithms, such as $\texttt{kNNG}_1$ and $\texttt{kNNG}_2$, MLE, and Hein, most of the times produce 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. Surprisingly DANCo is also able to provide correct estimate even on $\mathbf{M}_{\texttt{DSCV1}}$, which is known to have a fractal structure.

Furthermore, we compute the Friedman ranking test with the Bonferroni-Dunn post-hoc analysis to state the quality of the achieved results on both

Dataset	d	MLE	kNNG$_1$	kNNG$_2$	BPCA	Hein	CD	MiND$_{KL}$	DANCo	MLSVD
$\mathbf{M}_1$	*10.00*	9.10	9.16	9.89	5.45	9.45	9.12	10.30	10.09	**10.00**
$\mathbf{M}_2$	*3.00*	2.88	2.95	3.03	**3.00**	**3.00**	2.88	**3.00**	**3.00**	**3.00**
$\mathbf{M}_3$	*4.00*	3.83	3.75	3.82	**4.00**	**4.00**	3.23	**4.00**	**4.00**	2.08
$\mathbf{M}_4$	*4.00*	3.95	4.05	4.76	4.25	**4.00**	3.88	4.15	**4.00**	8.00
$\mathbf{M}_5$	*2.00*	1.97	1.96	2.06	**2.00**	**2.00**	1.98	**2.00**	**2.00**	**2.00**
$\mathbf{M}_6$	*6.00*	6.39	6.46	11.24	12.00	**5.95**	5.91	6.50	7.00	12.00
$\mathbf{M}_7$	*2.00*	1.96	1.97	2.09	**2.00**	**2.00**	1.93	2.07	**2.00**	2.35
$\mathbf{M}_9$	*20.00*	14.64	15.25	10.59	13.55	15.50	13.75	19.15	19.71	**20.00**
$\mathbf{M}_{10a}$	*10.00*	8.26	8.62	10.21	5.20	8.90	8.09	9.85	9.86	**10.00**
$\mathbf{M}_{10b}$	*17.00*	12.87	13.69	15.38	9.46	13.85	12.30	16.25	16.62	**17.00**
$\mathbf{M}_{10c}$	*24.00*	16.96	17.67	21.42	13.3	17.95	15.58	22.55	24.28	**24.00**
$\mathbf{M}_{10d}$	*70.00*	36.49	39.67	40.31	71.00	38.69	31.4	64.38	70.52	**70.00**
$\mathbf{M}_{11}$	*2.00*	2.21	1.95	2.03	1.55	2.00	2.19	**2.00**	**2.00**	1.00
$\mathbf{M}_{12}$	*20.00*	15.82	16.40	24.89	13.7	15.00	11.26	19.35	19.90	**20.00**
$\mathbf{M}_{13}$	*1.00*	**1.00**	0.97	1.07	5.70	**1.00**	1.14	**1.00**	**1.00**	**1.00**
$\mathbf{M}_{N1}$	*18.00*	12.25	14.26	19.8	36.00	14.10	10.40	17.76	18.76	**18.00**
$\mathbf{M}_{N2}$	*24.00*	14.72	17.62	26.87	48.00	17.76	12.43	23.76	25.76	**24.00**
$\mathbf{M}_{beta}$	*10.00*	6.36	6.45	14.77	19.7	4.00	3.05	7.00	7.00	**10.00**
$\mathbf{M}_{P3}$	*3.00*	2.89	2.93	3.12	7.00	2.00	2.43	**3.00**	**3.00**	1.00
$\mathbf{M}_{P6}$	*6.00*	4.96	4.98	5.82	7.00	2.66	3.58	5.04	**6.00**	1.00
$\mathbf{M}_{P9}$	*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 3.4: 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	i.d.	MLE	kNNG$_1$	kNNG$_2$	BPCA	Hein	CD	MiND$_{KL}$	DANCo	MLSVD
$\mathbf{M}_{\text{DSCV1}}$	*2.26*	2.03	1.77	1.86	6.00	3.00	1.92	2.50	**2.26**	1.75
$\mathbf{M}_{\text{ISOMAP}}$	*3.00*	4.05	3.60	4.32	4.00	**3.00**	3.37	3.9	4.00	1.00
$\mathbf{M}_{\text{SantaFe}}$	*9.00*	7.16	7.28	7.43	18.00	6.00	4.39	7.60	**8.19**	1.00
$\mathbf{M}_{\text{MNIST1}}$	*8.00-11.00*	10.29	10.37	9.58	11.00	8.00	6.96	11.00	9.98	1.00
$\mathbf{M}_{\text{Si02}}$	*12.00*	39.28	10.24	10.36	3.00	4.80	1.05	17.20	**12.60**	1.00
$\mathbf{M}_{\text{Isolet}}$	*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 3.5: 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 i.d. takes values in a range, we highlighted the results that best approximate the mean value of the range).

the synthetic and real datasets. table 3.6 and table 3.7 summarize the ranking results.

Finally, we performed the tests proposed in [65] to evaluate the robustness of MiND$_{KL}$, MLE, DANCo, and kNNG$_*$ w.r.t. the settings of their k param-

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 3.6: 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 3.7: Hypothesis testing of significance between techniques. Bonferroni-Dunn's procedure rejects those hypotheses that have a p-value$\leq$ 0.0125.

eter; these tests employ synthetic datasets sub-sampled from the standard Gaussian pdf in $\Re^5$ (i.d. $= 5$). We repeated the tests for datasets with cardinalities $N \in \{200, 500, 1000, 2000\}$ varying the parameter k in the range $\{5..100\}$.

As shown in figure 3.2 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. The reported results shown how i.d. estimators are strongly biased due to the effects related to high value of the i.d.

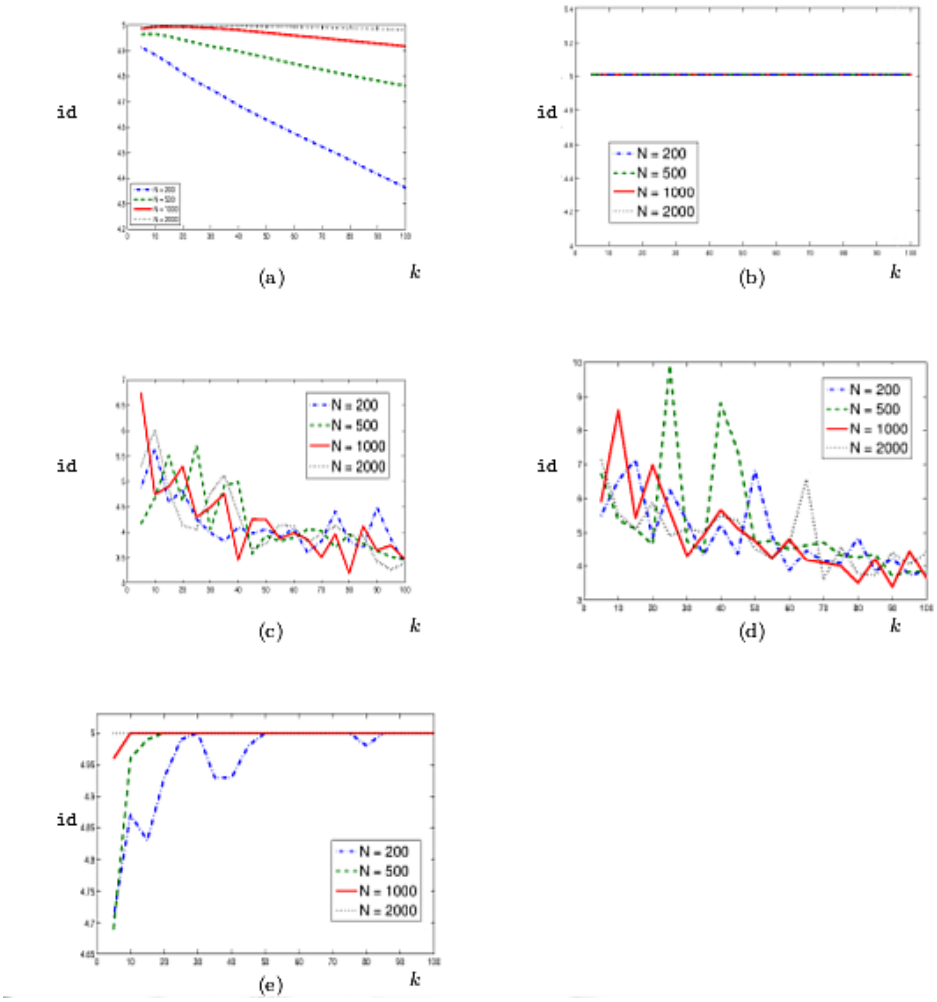

Figure 3.2: Behavior of: (a) `MLE`, (b) `DANCo` , (c) $\texttt{kNNG}_1$, (d) $\texttt{kNNG}_2$, and (e) $\texttt{MiND}_{\texttt{KL}}$ applied to points drawn from a 5-dimensional standard normal distribution; in this test $N \in \{200, 500, 1000, 2000\}$ and $k \in \{5..100\}$.

Chapter 4

Conclusion and Future Works

In this thesis we present novel estimators for the intrinsic dimension (`i.d.`) of a given dataset. This problem is a subject matter of many research efforts during the past decades, as we show in Chapter 1, where we survey the most notable `i.d.` estimators present in the literature, pointing out their main strength as well as their limitations and drawbacks. From this analysis it appeared that the best and most reliable techniques are the ones that are based on a local approach. Therefore, we develop algorithms based on the maximum likelihood of the distribution of the minimum neighbor distances of each local neighborhood, named $\texttt{MiND}_{\texttt{ML}*}$. Unfortunately, these techniques suffer from severe bias when facing high values of the `i.d.` (i.e. $d > 12$). To overcome these limitations we introduce a novel approach based on the use of Kullback-Leibler divergence between distance-related `pdf`s estimated locally both on the given dataset and on synthetic generated data of various dimensions ($d \in \{1, \cdots, D\}$). This approach forms the basis of a novel estimator, named $\texttt{MiND}_{\texttt{KL}}$, able to cope with high `i.d.` values. Even if $\texttt{MiND}_{\texttt{KL}}$ provides reliable results, it still encounters some difficulties when facing very high values of `i.d.` (i.e. $d > 30$). Consequently, we choose to add angle-related information to be exploited for `i.d.` estimation, due to the fact that the distribution of pairwise angles in a point neighborhood has proven to be more robust when used to estimate very high values of `i.d.`. This `i.d.` estimator, named `DANCo`, obtains more precise results, improving $\texttt{MiND}_{\texttt{KL}}$ through the addition of a jointly comparision of distance-related

distribution and pairwise angle-related distribution by means of Kullback-Leibler divergence.

Since `i.d.` estimators proposed in literature have been evaluated on different datasets, we propose a new standard benchmark framework to perform fair comparison between estimators in order to assess their quality. As far as we know, no benchmark framework existed in literature before, especially one entirely composed by public available datasets. To make comparison with our techniques possible, we also make publicly available[1] the code of $\texttt{MiND}_{\texttt{KL}}$ and `DANCo`, along with their optimized implementations.

The overall results show that methods based on the `pdf` comparison, namely $\texttt{MiND}_{\texttt{KL}}$ and `DANCo`, are promising and valuable techniques for `i.d.` estimation since they provide either the best `i.d.` estimates or values that are strongly comparable to the best ones. Moreover, these algorithms have shown to be robust in terms of their capability to: i) deal with both high and low `i.d.`s, ii) manage both linearly and nonlinearly embedded manifolds, and iii) deal with noisy datasets.

The strength of these methods relies in the use of the Kullback-Leibler divergences for the comparison of the `pdf`s related to norms and angles, locally estimated for each neighborhood both on the dataset and on generated hypersphere of growing dimensionality (i.e., $d \in \{1, \cdots, D\}$), being D the embedding dimensionality). Indeed, by means of the `pdf`s comparison it's possible to estimate a correct `i.d.` value overcoming the limitations of a local model based on neighborhood applied to an high dimensional space, such as the ones deriving from angle compression, norm compression, edge effect, and so on, as described in Chapter 1. It's worth noting that the exploitation of pairwise angles allows to use smaller neighborhoods, thus less prone to misrepresentation of the manifold structure, since the number of available samples is now equal to $\binom{k}{2}$, being k the number of neighbors. Finally, the derived closed form of the Kullback-Leibler divergences allow to use directly the distribution parameters estimated by means of maximum likelihood, instead of relying on samples that could be affected by noise.

During the development of these techniques, we also laid the foundations for further research work, with the aim to improve and extend these proposed estimators.

[1] http://www.mathworks.it/matlabcentral/fileexchange/40112-intrinsic-dimensionality-estimation-techniques

In the previous chapter, empirical results shown that `DANCo` is able to provide a fractal value of the `i.d.` as the mean of several runs on the same dataset; further research work will be dedicated to formalize the capability to deal with non integer value of the `i.d.`, investigate the relations with the definitions of fractal dimension stated in Chapter 1 and eventually providing a bound on the number of runs needed to obtain an unbiased result of a fractal dimension.

We also noted that employing finite sets of data drawn from strongly non-uniform `pdf` could reduce the performance of our estimators as well as other state-of-the-art techniques. For this reason further studies in this direction will be part of our future works.

Moreover, a major improvement, that is under current research, is the relaxation of the single manifold assumption: our future research work is devoted to an extension or modification of these techniques, in order to cope with dataset sampled from more than one manifold, still providing a reliable estimate of the `i.d.` for each of them.

Finally, most of the novel and advanced state-of-the-art techniques for dimensionality reduction by means of manifold learning are based on the construction of local neighborhoods. Unfortunately, in general these methods don't consider a measure for assessing the quality of the generated neighborhoods, and the lack of this preliminary step could lead in a reduction of dimension that doesn't preserve the manifold structure. For this reason, we are currently testing the local information exploited in our techniques as a measure of how the neighborhood matches the manifold structure. We believe that we'll obtain promising results on this task, since we observed improvements in terms of clustering accuracy using these informations as features [15].

Appendix A: Implementations

In this appendix the pseudo-codes of the presented algorithms are reported. We show here the psuedo-code of $\texttt{MiND}_{\texttt{MLi}}$, $\texttt{MiND}_{\texttt{KL}}$, `DANCo` and its optimization `FastDANCo`. In the following pseudo-codes $kNN(\mathbf{X}_N, \mathbf{x}, k)$ is the procedure that employs a k-nearest neighbor search returning the set of the k nearest neighbors of $\mathbf{x}$ in $\mathbf{X}_N$, whereas in the pseudo-code for `FastDANCo` $[s^k_{\check{d}_{d,ML}}, s^k_{\check{\mu}^d_\nu}, s^k_{\check{\mu}^d_\tau}] : \mathbb{N}^2_+ \to \mathbb{R}^3_+$ are the fitting functions that, given the dimensionality d of the uniform hypersphere and the cardinality N of the vectors sampled from it, computes the values of $[\check{d}_{d,ML}, \check{\mu}^d_\nu, \check{\mu}^d_\tau]$ used for estimating the `i.d.`.

Listing 4.1: Pseudocode for the $\texttt{MiND}_{\texttt{MLi}}$ algorithm.

Input:
  $\mathbf{X}_N$: The dataset points $\{\mathbf{x}_i\}_{i=1}^N$.
  k: The kNN parameter.
Output:
  $\hat{d}$: The estimated intrinsic dimensionality.
{`Compute for each point the normalized radii`}
`for i:=1 to N do begin`
  $\bar{\mathbf{X}}_{k+1} = kNN(\mathbf{X}_N, \mathbf{x}_i, k)$; {`Finding the` k `neighbors of` $\mathbf{x}_i$.}
  $\rho(\mathbf{x}_i) = \min_{\mathbf{x}_j \in \bar{\mathbf{X}}_{k+1}} \|\mathbf{x}_i - \mathbf{x}_j\| / \max_{\hat{\mathbf{x}} \in \bar{\mathbf{X}}_{k+1}} \|\mathbf{x}_i - \hat{\mathbf{x}}\|$;
`end`
{`Choosing` $\hat{d} \in \{1..D\}$ `that maximizes the log likelihood`}
$\hat{d} = \arg\max_{d \in \{1..D\}} \left((d-1) \sum_{\mathbf{x}_i \in \mathbf{X}_N} \log \rho(\mathbf{x}_i) + (k-1) \sum_{\mathbf{x}_i \in \mathbf{X}_N} \log \left(1 - \rho^d(\mathbf{x}_i)\right)\right)$;

Listing 4.2: Pseudocode for the $\texttt{MiND}_{\texttt{KL}}$ algorithm.

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

Listing 4.3: Pseudocode for the `DANCo` algorithm.

Input:
  $\mathbf{X}_N$: The dataset points $\{\mathbf{x}_i\}_{i=1}^N$.
  k: The `kNN` parameter.
Output:
  $\hat{d}$: The estimated intrinsic dimensionality.

  {`Compute for each point the normalized radii and the pairwise angles`}
  $[\hat{\mathbf{r}}, \hat{\nu}, \hat{\tau}] = AngleNormInfo(\mathbf{X}_N, k)$;

  {`Choose` $d_{ML} \in [1, D]$ `that maximizes the log likelihood`}
  $\hat{d}_{ML} = \arg\max_{1 \leq d \leq D} \left(N \log kd + (d-1) \sum_{i=1}^N \log \hat{r}_i + (k-1) \sum_{i=1}^N \log \left(1 - \hat{r}_i^d\right) \right)$;
  `if` $(\hat{d}_{ML} < 2)$; $\hat{d} = \text{round}(\hat{d}_{ML})$; `return`;

  {`Average the VM parameters`}
  $\hat{\mu}_\nu = \text{atan}_2(\sum_{i=1}^N \sin \hat{\nu}_i, \sum_{i=1}^N \cos \hat{\nu}_i)$;
  $\hat{\mu}_\tau = N^{-1} \sum_{j=1}^N \hat{\tau}_j$;

  `for d=1:D`
    {`Uniformly sample from the unit ball obtaining a hs`^{d}`-sample`}
    $\mathbf{Y}_{Nd} = \{\mathbf{y}_i = \bar{\mathbf{y}} / \|\bar{\mathbf{y}}\| u^{1/d}; \bar{\mathbf{y}} \sim \mathcal{N}(\cdot | \mathbf{0}_d, 1), u \sim U(0,1)\}_{i=1}^N$;

    {`Compute for each point the normalized radii and the pairwise angles`}
    $[\check{\mathbf{r}}^d, \check{\nu}^d, \check{\tau}^d] = AngleNormInfo(\mathbf{Y}_{Nd}, k)$;

    {`Choose` $d_{ML} \in [1, D]$ `that maximizes the log likelihood`}
    $\check{d}_{d,ML} = \arg\max_{1 \leq d \leq D} \left(N \log kd + (d-1) \sum_{i=1}^N \log \check{r}_i + (k-1) \sum_{i=1}^N \log \left(1 - \check{r}_i^d\right) \right)$;

    {`Average the VM parameters`}
    $\check{\mu}_\nu^d = \text{atan}_2(\sum_{i=1}^N \sin \check{\nu}_i, \sum_{i=1}^N \cos \check{\nu}_i)$;
    $\check{\mu}_\tau^d = N^{-1} \sum_{i=1}^N \check{\tau}_i$;
  `end`

  {`Estimate the intrinsic dimensionality`}
  $\hat{d} = \arg\min_{d \in \{1..D\}} \mathcal{KL}(g(\cdot; k, \hat{d}_{ML}), g(\cdot; k, \check{d}_{d,ML})) + \mathcal{KL}(q(\cdot; \hat{\mu}_\nu, \hat{\mu}_\tau), q(\cdot; \check{\mu}_\nu^d, \check{\mu}_\tau^d))$;

Listing 4.4: Pseudocode for the `FastDANCo` algorithm.

Input:

$\mathbf{X}_N$: The dataset points $\{\mathbf{x}_i\}_{i=1}^N$.

k: The `kNN` parameter.

$[s^k_{\check{d}_{d,ML}}, s^k_{\check{\mu}^d_\nu}, s^k_{\check{\mu}^d_\tau}] : \mathbb{N}^2_+ \to \mathbb{R}^3_+$: The functions fitting $\mathtt{hs}^d$-samples, i.e. $[\check{\mathbf{d}}_{ML}, \check{\nu}, \check{\tau}]$ w.r.t. the dimensionality d and the sample size N.

Output:

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

{`Compute for each point the normalized radii and the pairwise angles`}

$[\hat{\mathbf{r}}, \hat{\nu}, \hat{\tau}] = AngleNormInfo(\mathbf{X}_N, k)$;

{`Choose` $d_{ML} \in [1, D]$ `that maximizes the log likelihood`}

$\hat{d}_{ML} = \arg\max_{1 \le d \le D} \left(N \log kd + (d-1) \sum_{i=1}^N \log \hat{r}_i + (k-1) \sum_{i=1}^N \log \left(1 - \hat{r}_i^d\right) \right)$;

`if` $(\hat{d}_{ML} < 2)$; $\hat{d} = \text{round}(\hat{d}_{ML})$; `return`;

{`Average the VM parameters`}

$\hat{\mu}^d_\nu = \text{atan}_2(\sum_{i=1}^N \sin \hat{\nu}_i, \sum_{i=1}^N \cos \hat{\nu}_i)$;

$\hat{\mu}_\tau = N^{-1} \sum_{j=1}^N \hat{\tau}_j$;

`for d=1:D`

{`Invoke the fitting functions to obtain` d_{ML} `and the VM parameters`}

$\check{d}_{d,ML} = s^k_{\check{d}_{d,ML}}(N, d)$;

$\check{\mu}^d_\nu = s^k_{\check{\mu}^d_\nu}(N, d)$;

$\check{\mu}^d_\tau = s^k_{\check{\mu}^d_\tau}(N, d)$;

`end`

{`Estimate the intrinsic dimensionality`}

$\hat{d} = \arg\min_{d \in \{1..D\}} \mathcal{KL}(g(\cdot; k, \hat{d}_{ML}), g(\cdot; k, \check{d}_{d,ML})) + \mathcal{KL}(q(\cdot; \hat{\mu}_\nu, \hat{\mu}_\tau), q(\cdot; \check{\mu}^d_\nu, \check{\mu}^d_\tau))$;

Listing 4.5: Pseudocode for the function *AngleNormInfo*.

Input:
  $\mathbf{X}_N$: The dataset points $\{\mathbf{x}_i\}_{i=1}^N$.
  k: The `kNN` parameter.
Output:
  $\mathbf{r}$: The normalized distance.
  ν: The mean angle of a `VM` distribution.
  τ: The concentration paramater of a `VM` distribution.

  {`Compute for each point the normalized radii and the pairwise angles`}
  `for i=1:N`
    {`Compute the normalized distances`}
    $\bar{\mathbf{X}}_{k+1} = kNN(\mathbf{X}_N, \mathbf{x}_i, k)$; {`Find the` k `neighbors of` $\mathbf{x}_i$ `in` $\mathbf{X}_N$.}
    $\mathbf{r}_i = \rho(\mathbf{x}_i) = \min_{\mathbf{x}_j \in \bar{\mathbf{X}}_{k+1}} \|\mathbf{x}_i - \mathbf{x}_j\| / \max_{\mathbf{x} \in \bar{\mathbf{X}}_{k+1}} \|\mathbf{x}_i - \mathbf{x}\|$;

    {`Compute the pairwise Angles`}
    $T = 1$;
    `for j=1:k`
      $\mathbf{x}_j = \mathbf{x}_j - \mathbf{x}_i$;
      `for z=j+1:k`
        $\theta_T = \arccos \frac{\mathbf{x}_z \cdot \mathbf{x}_j}{\|\mathbf{x}_z\| \|\mathbf{x}_j\|}$;
        $T = T + 1$;
      `end`
    `end`

    {`Calculate the VM parameters that maximizes the log likelihood`}
    $\nu_i = \operatorname{atan}_2\left(\sum_{j=1}^T \sin\theta_j, \sum_{j=1}^T \cos\theta_j\right)$;
    $\eta = \sqrt{\left(\frac{1}{T}\sum_{j=1}^T \cos\theta_j\right)^2 + \left(\frac{1}{T}\sum_{j=1}^T \sin\theta_j\right)^2}$;
    `if` $(\eta < 0.53)$; $\tau_i = 2\eta + \eta^3 + \frac{5\eta^5}{6}$; `end`;
    `elseif` $((0.53 \leq \eta)\&\&(\eta < 0.85))$; $\tau_i = -0.4 + 1.39\eta + \frac{0.43}{1-\eta}$; `end`;
    `else` $\tau_i = \frac{1}{\eta^3 - 4\eta^2 + 3\eta}$; `end`;
  `end`

Bibliography

[1] M. Abramowitz and I. A. Stegun. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. Dover, New York, ninth Dover printing, tenth GPO printing edition, 1964.

[2] Y. Ashkenazy. The use of generalized information dimension in measuring fractal dimension of time series. *Physica A: Statistical Mechanics and its Applications*, 271(3–4):427 – 447, 1999.

[3] 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.

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

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

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

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

[8] 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.

[9] E. Breitenberger. Analogues of the normal distribution on the circle and the sphere. *Biometrika*, 50, 1963.

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

[13] 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.

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

[15] P. Campadelli, E. Casiraghi, C. Ceruti, G. Lombardi, and A. Rozza. Local intrinsic dimensionality based features for clustering. In *Image Analysis and Processing–ICIAP 2013*, pages 41–50. Springer, 2013.

[16] 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.

[17] 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.

[18] 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*, 47(8):2569–2581, 2014.

[19] J. Chen, H.R. Fang, and Y. Saad. Fast approximate kNN graph construction for high dimensional data via recursive lanczos bisection. *J. of Machine Learning Research*, 10:1989–2012, 2009.

[20] 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.

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

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

[23] T. F. Coleman and Y. Li. An interior, trust region approach for nonlinear minimization subject to bounds. *SIAM J. on Optimization*, 6:418–445, 1996.

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

[25] 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.

[26] 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.

[27] 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.

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

[29] 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.

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

[31] 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.

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

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

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

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

[36] N. I. Fisher. *Statistical Analysis of Circular Data*. Cambridge University Press, January 1996.

[37] G. S. Fishman. *Monte Carlo: Concepts, Algorithms, and Applications*. Springer Series in Operations Research. Springer-Verlag, New York, NY, 1996.

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

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

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

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

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

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

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

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

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

[47] R. Heylen and P. Scheunders. Hyperspectral Intrinsic Dimensionality Estimation with Nearest-Neighbor Distance Ratios. *IEEE Journal of Selected Topics in Applied Eearth Observations and Remote Sensing*, 6(2):570–579, 2013.

[48] M. Hilbert and P. López. The world's technological capacity to store, communicate, and compute information. *Science*, 332(6025):60–65, 2011.

[49] G. W. Hill. New approximations to the von Mises distribution. *Biometrika*, 63(3):673–676, 1976.

[50] 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.

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

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

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

[54] 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.

[55] R. Karbauskaite, G. Dzemyda, and E. Mazetis. Geodesic distances in the maximum likelihood estimator of intrinsic dimensionality. *Non linear Analysis: Modelling and Control*, 16:387–402, 2011.

[56] D.N. Kaslovsky and F.G. Meyer. Optimal Tangent Plane Recovery From Noisy Manifold Samples. *CoRR–arXiv*, 2011.

[57] 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.

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

[59] 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.

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

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

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

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

[64] S. R. Lehky, R. Kiani, H. Esteky, and K. Tanaka. Dimensionality of object representations in monkey inferotemporal cortex. 2014.

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

[66] 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.

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

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

[69] 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.

[70] 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.

[71] G. Lombardi, A. Rozza, C. Ceruti, E. Casiraghi, and P. Campadelli. Minimum neighbor distance estimators of intrinsic dimension. In *Machine Learning and Knowledge Discovery in Databases*, pages 374–389. Springer, 2011.

[72] R. D. Lord. The use of the Hankel transform in statistics I. general theory and examples. *Biometrika*, 41(1/2):44–55, 1954.

[73] 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/.

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

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

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

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

[78] B. O'Neill. *Elementary Differential Geometry.* Elsevier Ac. Press, 2006.

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

[80] R. Paredes, E. Chávez, K. Figueroa, and G. Navarro. Practical construction of k-nearest neighbor graphs in metric spaces. In C. Àlvarez and M.J. Serna, editors, *WEA*, volume 4007 of *Lecture Notes in Computer Science*, pages 85–97. Springer, 2006.

[81] 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.

[82] 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.

[83] 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.

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

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

[86] 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.

[87] J.C. Robinson. *Dimensions, Embeddings, and Attractors.* Cambridge Tracts In Mathematics. Cambridge University Press, 2010.

[88] A. Rozza, G. Lombardi, C. Ceruti, E. Casiraghi, and P. Campadelli. Novel high intrinsic dimensionality estimators. *Machine Learning Journal*, May 2012.

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

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

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

[92] A. Sodergren. On the distribution of angles between the N shortest vectors in a random lattice. *J. London Math. Soc.*, 84(3):749–764, 2011.

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

[94] 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.

[95] 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.

[96] 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.

[97] 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.

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

[99] 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.

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

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

[102] G. J. G. Upton. New approximations to the distribution of certain angular statistics. *Biometrika*, 61(2):369–373, 1974.

[103] G. J. G. Upton. Approximate confidence intervals for the mean direction of a von Mises distribution. *Biometrika*, 73(2):525–527, 1986.

[104] 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.

[105] L.J.P. van der Maaten, E.O. Postma, and H.J. van den Herik. Dimensionality reduction: A comparative review. *Journal of Machine Learning Research*, 10(1-41):66–71, 2009.

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

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

[108] A. P. N. Vo, S. Oraintara, and T. T. Nguyen. Statistical image modeling using von Mises distribution in the complex directional wavelet domain. In *Proc. of ISCAS 2008*, pages 2885–2888, 2008.

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

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

[111] P.L. Zador. Asymptotic quantization error of continuous signals and the quantization dimension. *IEEE Trans. Information Theory*, IT-28:139–148, 1982.

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