Pattern Recognition 47 (2014) 2569–2581

Contents lists available at ScienceDirect

Pattern Recognition

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

DANCo: An intrinsic dimensionality estimator exploiting angle and norm concentration

Claudio Ceruti [a,*], Simone Bassis [c], Alessandro Rozza [b], Gabriele Lombardi [c], Elena Casiraghi [c], Paola Campadelli [c]

[a] Dipartimento di Matematica, Università degli Studi di Milano, Via Saldini 50, Milan, Italy
[b] Dipartimento di Scienze e Tecnologie, Università degli Studi di Napoli – Parthenope, Centro Direzionale, Isola C4, Naples, Italy
[c] Dipartimento di Informatica, Università degli Studi di Milano, Via Comelico 39/41, Milan, Italy

ARTICLE INFO

Article history:
Received 13 July 2013
Received in revised form
24 November 2013
Accepted 19 February 2014
Available online 5 March 2014

Keywords:
Intrinsic dimensionality estimation
Manifold learning
Von Mises distribution
Nearest neighbor distance distribution
Kullback–Leibler divergence

ABSTRACT

In the past decade the development of automatic intrinsic dimensionality estimators has gained considerable attention due to its relevance in several application fields. However, most of the proposed solutions prove to be not robust on noisy datasets, and provide unreliable results when the intrinsic dimensionality of the input dataset is high and the manifold where the points are assumed to lie is nonlinearly embedded in a higher dimensional space. In this paper we propose a novel intrinsic dimensionality estimator (DANCo) and its faster variant (FastDANCo), which exploit the information conveyed both by the normalized nearest neighbor distances and by the angles computed on couples of neighboring points. The effectiveness and robustness of the proposed algorithms are assessed by experiments on synthetic and real datasets, by the comparative evaluation with state-of-the-art methodologies, and by significance tests.

1. Introduction

Given a dataset $\boldsymbol{X}_N \equiv \{\boldsymbol{x}_i\}_{i=1}^{N} \subset \Re^D$, its intrinsic dimensionality (id) is generally defined as the minimum number of parameters needed to represent the data without information loss. In the last decade a great deal of research work has been devoted to the development of id estimation algorithms; to this aim, the feature vectors $\boldsymbol{x}_i$ are generally viewed as points constrained to lie on a low dimensional manifold $\mathcal{M} \subseteq \Re^d$ (generally assumed to be smooth or locally smooth) embedded in a higher dimensional space $\Re^D$, where the manifold dimensionality d is the id to be estimated. To generalize this assumption, Fukunaga [1] defines the id of $\boldsymbol{X}_N$ as the dimensionality $d \in \{1 \ldots D\}$ of the subspace of $\Re^D$ where all the points in $\boldsymbol{X}_N$ entirely lie.

The id is a very useful information for several reasons; first, dimensionality reduction techniques, often used to reduce the "curse of dimensionality" effect [2] by computing a more compact representation of the data, are profitable when the number of projection dimensions is the minimal one (namely the id) that allows us to retain the maximum amount of useful information expressed by the data. Moreover, when using auto-associative neural networks for nonlinear feature extraction, the id can suggest a reasonable value for the number of hidden neurons [3]. Besides, according to statistical learning theory [4], the capacity and the generalization capability of a classifier may depend on the id; indeed, Friedman et al. in [5] mark that, in order to balance a classifier's generalization ability and its empirical error, the complexity of the classification model should be related to the id of the available dataset. Finally, id estimates are particularly useful for several practical applications. Two interesting examples are described in [6,7]. More precisely, Camastra and Filippone in [6] show that id estimation methods can be successfully used to evaluate the model order in a time series, which is crucial to make reliable time series predictions; this consideration is supported by the fact that the domain of attraction of a nonlinear dynamic system has a very complex geometric structure and the studies on the geometry of the attraction domain are closely related to fractal geometry, and therefore to fractal dimension (which can be approximated by the id estimate). The second example [7] has been recently published in the field of crystallography to show that the id is a useful information when analyzing crystal structures. This study not only proves that the knowledge of the id is especially useful when dealing with practical tasks concerning real data, but also underlines the need to manage datasets drawn from manifolds characterized by high id and embedded in spaces of much greater dimensionality. Due to these facts, the development

[*] Corresponding author. Tel.: +39 250316285; fax: +39 250316334.
E-mail addresses: claudio.ceruti@unimi.it (C. Ceruti), bassis@di.unimi.it (S. Bassis), alessandro.rozza@uniparthenope.it (A. Rozza), lombardi@di.unimi.it (G. Lombardi), casiraghi@di.unimi.it (E. Casiraghi), campadelli@di.unimi.it (P. Campadelli).

http://dx.doi.org/10.1016/j.patcog.2014.02.013
0031-3203 © 2014 Elsevier Ltd. All rights reserved.

of effective estimators coping with high id datasets has gained increasing importance.

Unfortunately, even if a great deal of research work has been focused on the development of id estimators, and several interesting techniques have been presented in the literature, to our knowledge only few methods [8–11] have investigated the problem of input datasets having a sufficiently high id (namely id ≥ 10) and being drawn from manifolds nonlinearly embedded in higher dimensional spaces; indeed, when the id of this kind of data must be estimated, several well-known techniques compute underestimates (see Section 5). These considerations lead us to the development of an id estimator, called "DANCo" (Dimensionality from Angle and Norm Concentration) whose effectiveness is shown by experiments on both synthetic and real datasets, by the comparative evaluation with relevant state-of-the-art algorithms, and by significance tests. The peculiarity and strength of the proposed estimator is to be sought in the joint usage of normalized nearest neighbor distances and mutual angles; indeed, as further confirmed by our experimental analysis and theoretical considerations, the coupled exploitation of the distributions of normalized nearest neighbor distances and mutual angles increases the id estimation accuracy when dealing with both low and high-dimensional manifolds.

Based on this consideration, in this paper we first describe how we could compute an id estimate by solely employing either normalized nearest neighbor distances or mutual angles, and then we describe how they can be combined to obtain the final, and more reliable, id estimation. Furthermore, we propose Fast-DANCo, a faster variant of DANCo that achieves a comparable id estimation accuracy.

The paper is organized as follows: in Section 2 state-of-the-art id estimators are reviewed. In Section 3 base theoretical considerations laying foundations for the proposed estimators are presented. In Section 4 we describe DANCo and FastDANCo, and we analyze their properties. In Section 5 robustness w.r.t. noise and parameter settings, and a detailed comparison with state-of-the-art methodologies on a wide family of public datasets are supported by significance tests. Section 6 reports conclusions.

2. Related works

In this section we recall relevant state-of-the-art id estimators. A detailed description of id estimators published before 2003 can also be found in the extensive survey of Camastra [12].

The most cited example of id estimator is Principal Component Analysis, PCA [13]; this well-known technique is often used as the first step of several machine learning methods to perform dimensionality reduction, whose purpose is to map a set of high dimensional data into a low dimensional space while preserving the intrinsic structure of the data [14]. Given a dataset of observed points in $\Re^D$ and assuming that all the information related to the intrinsic structure is expressed by the data variance, PCA computes their low dimensional representations in $\Re^d$ by projecting the points in $\Re^D$ on the directions of their maximum variance, also called "principal components" (PCs). Applying PCA, the number of retained PCs is viewed as the id estimate; its drawbacks are the difficult choice of the threshold used to select the number of PCs to be retained, and its incapability to deal with manifolds nonlinearly embedded in higher dimensional spaces.

To cope with the last problem, Fukunaga [15] achieves more accurate results by applying a local PCA that works in small subregions of the dataset to estimate their local ids; all the local estimates are then combined to compute the id estimate characterizing the whole dataset. Unfortunately, the correct selection of the local regions and thresholds could be difficult, as shown by the empirical evaluations reported in [16].

Tipping and Bishop [17] noted that PCA and its variants are deterministic methodologies lacking an associated probabilistic model for the observed data and a method for selecting the number of PCs to be retained. For this reason, they present the Probabilistic PCA (PPCA), by reformulating PCA as the maximum likelihood solution of a specific latent variable model. The latent space dimensionality $\hat{d}$ is considered as the id estimate. PPCA has been successfully applied to problems in data compression, density estimation and data visualization, and has been extended to both mixture and hierarchical mixture models, but it still does not provide any mechanism for estimating the best value of the latent space dimensionality d, which corresponds to the id estimate [18]. For this reason, Bishop [18] extends the PPCA model by defining a Bayesian treatment of PCA (Bayesian PCA or BPCA).

Though BPCA is a theoretically founded approach for id estimation, it is based on a strong assumption of normally distributed data. Therefore, Li and Tao [19] have recently proposed a Simple Exponential Family PCA (SePCA) that is a generalized version of probabilistic PCA. SePCA extends BPCA by substituting the Gaussian distributions with exponential family distributions. Despite the promising results reported by the authors, the performance of SePCA is highly dependent on the parameter setting and its application is successful only if the data distribution is known, which is often not the case.

Other two interesting works improving PCA have been reported in [20,21]. Precisely, the Sparse Principal Component Analysis (SPCA, [20]) is obtained by reformulating PCA as a regression optimization problem and imposing the lasso constraint on the regression coefficients. Since SPCA requires the manual setting of the constraint weights, in [21] the authors introduced the Sparse Probabilistic Principal Component Analysis (SPPCA), that is a probabilistic Bayesian formulation of SPCA that uses a different prior to achieve sparsity. In this way, SPPCA can automatically learn the hyper-parameter related to the weight of the constraint of SPCA through a type II maximum likelihood.

The aforementioned PCA-based methods are generally classified as projection methods [12,22] since they search for the best subspace to project the data. Unfortunately, though these methods have shown to be useful for several applications, they cannot provide reliable id estimates since they are too sensitive to noise and parameter settings [22].

Other id estimators, such as Nearest Neighbor estimator [23], and Tensor Voting Framework (TVF, [24]), which are generally classified as geometric methods [22], exploit the intrinsic geometry of the dataset and are often relying on the consideration that the volume of a d-dimensional set scales with its size r as r^d, which implies that also the number of samples covered by a hypersphere with radius r grows proportionally to r^d. Based on this consideration, most of the geometric methods consider hyperspheres with sufficiently small radius r and centered on the points in the dataset, and exploit either their fractal dimension estimate or statistics related to the distances between nearest neighbors in the hypersphere or between the hypersphere center and the points it contains; both the fractal dimensions and the aforementioned statistics are expressed as functions of the id of the manifold from which the points have been randomly drawn.

Perhaps the most popular geometric estimator, based on the aforementioned consideration of dependency between r and d, is the Correlation Dimension (CD, [25]).

Noticing that the assumption exploited by CD is true only when considering a range of values around the proper (i.e. locally linear) scale r, Brand in [26] proposes an approach (hereinafter *Charting Manifold*) that simultaneously estimates r and the id by looking at how r-balls centered on the data intercept points as r grows.

Another geometric estimator, referred to as Hein in the following [27], exploits the asymptotes of a smoothed version of the CD estimate to tackle the dependency of the CD estimator from the scale r.

Considering the same assumption of CD and trying to avoid its dependency from the choice of the proper scale r, in [28] the authors propose an interesting approach that estimates the id of a manifold in a small neighborhood of a selected point, and analyzes its finite-sample convergence properties.

A well-known approach, derived by considering the theories at the basis of the fractal dimension, is the Packing Number technique [29]. It exploits the r-packing number $M(r)$ of the dataset $\boldsymbol{X}_N \subset \mathcal{S}$, where $\mathcal{S}$ is a metric space with distance metric $\delta(\cdot,\cdot)$. More precisely, $\boldsymbol{X}_N$ is said to be r-separated if $\forall \boldsymbol{x}, \boldsymbol{y} \in \boldsymbol{X}_N, \boldsymbol{x} \neq \boldsymbol{y} \Rightarrow \delta(\boldsymbol{x}, \boldsymbol{y}) \geq r$, and $M(r)$ is the maximum cardinality of an r-separated subset of $\boldsymbol{X}_N$. The authors demonstrate a relation between d and $M(r)$, and then derive an expression to compute the id estimate.

The geometric estimator introduced in [30] (hereinafter *Quantization method*) is based on the theoretical analysis of a k-vector quantizer applied to the D-dimensional dataset $\boldsymbol{X}_N \subset \mathcal{M} \subseteq \Re^d$ to partition it into k subclasses. The id estimate is computed by exploiting the connection [31] between the id of a smooth manifold $\mathcal{M}$ and the asymptotic optimal k-point quantization error, that is the distortion introduced by the optimal k-vector quantizer for $\boldsymbol{X}_N$. This method is equivalent to the Packing Number technique when $k \to 0$.

The Maximum Likelihood Estimator for intrinsic dimensionality (MLE, [22]) also analyzes point neighborhoods, by applying the principle of maximum likelihood to the distances between close neighbors, and deriving the estimator by a Poisson process approximation.

In [32] the authors propose an algorithm that exploits entropic graphs to estimate both the id of a manifold and the intrinsic entropy of the manifold random samples. This technique is based on the observation that the length function of such graphs, that is the sum of arc weights on the minimal graph that spans all the points in the dataset, is strongly dependent on d. The authors test their method by adopting either the Geodesic Minimal Spanning Tree (GMST, [33]) where the arc weights are the geodetic distances computed by the ISOMAP algorithm [34], or the kNN-Graph (kNNG, [32]), where the arc weights are based on the Euclidean distances, which require a lower computational cost. As mentioned before, it is undoubted that many neighborhood based estimators usually underestimate the id when its value is sufficiently high and, to our knowledge, only few works address this problem [8–11]. Indeed, as shown in [35], the number of sample points required to perform dimensionality estimation grows exponentially with the id "curse of dimensionality". More precisely, the authors prove that the CD algorithm provides accurate id estimates only if the dataset cardinality is $N > 10^{d/2}$. Therefore, when the dimensionality is too high the number of sample points practically available is insufficient to compute a reliable id estimate. Moreover, as a result of the so-called "edge effect" phenomenon [16], the ratio between the points close to the edge of the manifold and the points inside it raises in probability when the dimensionality increases, affecting the results achieved by estimators based on statistics related to the behavior of point neighborhoods.

To overcome these difficulties, in [8] the authors propose an empirical id correction procedure based on the estimation of the error obtained on synthetically produced datasets of known dimensionality. More precisely, after generating D datasets characterized by incremental id values ($d_i \in \{1..D\}$), the authors apply the CD algorithm [25] to estimate the id ($\hat{d}_i$) of each dataset. Fitting the points ($d_i, \hat{d}_i$) they obtain the so-called "correction curve" used to adjust the id estimates. In [10] the authors propose IDEA, a geometric estimator based on an asymptotic correction technique. Given a dataset of unknown id, this technique extracts random subsets of different cardinalities and computes their id estimates; subsequently, the bidimensional points composed by the cardinality of each subset and by its id estimate are fitted with a curve having a horizontal asymptote whose ordinate is the id estimate. The last id estimator we recall is MiND$_{\text{KL}}$ [9]; it is based on the comparison between the empirical probability density function (pdf) of the neighborhood distances computed on the dataset and the distribution of the neighborhood distances computed from points uniformly drawn from hyperspheres of known increasing dimensionality; the id estimate is the value that minimizes their Kullback–Leibler divergence (KL).

In Table 1 all the surveyed methods are summarized, according to their categories and description.

Table 1
Summary table of the surveyed id estimators.

Category	Name	Description
Projection method	PCA	Maximum variance approach.
	Local PCA	Local version of PCA.
	PPCA	Probabilistic treatment of PCA.
	BPCA	Bayesian treatment of PCA.
	SePCA	Generalization of PPCA.
	SPCA	Reformulation of PCA as a regression optimization problem.
Geometric method	*Nearest neighbor estimator*	id estimator based on the distance between neighbors.
	TVF	id estimator based on the Tensor Voting Framework.
	CD	Correlation dimension estimator.
	Charting manifold	Improvement of CD.
	Hein	Asymptotic smooth version of CD.
	Packing number	r-packing number estimator.
	MLE	Method applying the maximum likelihood to the distance between neighbors.
	Quantization method	Asymptotic optimal k-point quantization error estimator.
	GMST	Entropic graph technique (geodetic distances).
	kNNG	Entropic graph technique (euclidean distances).
Correction method	IDEA	Asymptotic correction method.
	MiND$_{\text{KL}}$	Correction method based on the comparison between pdfs.

3. Theoretical considerations

In this section, before introducing theoretical considerations specifically concerning nearest neighbor distances (see Section 3.1) and mutual angles (see Section 3.2), we recall a theorem introduced in [9] to justify the use of local information to estimate global properties of the manifold where the points lie.

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

To face this problem, first consider the specific case of a manifold $\mathcal{M} \equiv \mathcal{B}_d(\boldsymbol{0}_d, 1) \subset \Re^d$, where $\mathcal{B}_d(\boldsymbol{0}_d, 1)$ is the unit hypersphere centered in the origin and uniformly sampled; furthermore, assume the embedding map ϕ to be the identity map. Considering k points $\{\boldsymbol{z}_i\}_{i=1}^k$ uniformly drawn from $\mathcal{B}_d(\boldsymbol{0}_d, 1)$, our aim is to find the pdf related to the minimum distance between the k points and the hypersphere center $\boldsymbol{0}_d$.

To this aim, considering a generic point $\boldsymbol{z}_i, i \in 1, \ldots, k$, we denote with $p(r)$ the pdf for the event $\|\boldsymbol{z}_i\| = r$ ($r \in [0, 1]$) where $\|\cdot\|$ is the

L_2 norm operator, and with $P(\check{r}<r)$ the probability for the event $\|\boldsymbol{z}_i\|<r$. Being $\boldsymbol{z}_i$ uniformly drawn it is possible to evaluate $P(\check{r}<r)$ by means of hypersphere volume ratios, that is by computing the volume, V_r, of a d-dimensional hypersphere of radius r, and normalizing it by the volume, V_1, of the unit d-dimensional hypersphere. V_r is computed as follows:

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

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

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

$$g(r;d,k) \propto \breve{g}(r;d,k) = p(r)(1-P(\check{r}<r))^{k-1}$$

$$= \frac{\partial\left(\frac{V_r}{V_1}\right)}{\partial r}\left(1-\frac{V_r}{V_1}\right)^{k-1} = \frac{1}{V_1} d r^{d-1}(1-r^d)^{k-1}$$

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

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

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

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

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

We are now ready to prove Theorem 1, whose statement is reported below for the sake of completeness.

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

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

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

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

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

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

Defining

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

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

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

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

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

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

According to these observations, our technique exploits the statistical properties of norms and mutual angles computed on points uniformly drawn from unit hyperspheres; to this aim, in Sections 3.1, 3.2 we analyze the statistical properties of norms and angles respectively, while in Section 3.3 we describe how these properties can be simultaneously used to define id estimators.

3.1. Concentration of norms

Consider the problem of estimating the id of a manifold $\mathcal{M} \equiv \Re^d$ embedded in a higher dimensional space $\Re^D$ by means of a proper locally isometric smooth map $\phi: \mathcal{M} \to \Re^D$, by employing samples drawn from $\mathcal{M}$ through a smooth pdf $f: \mathcal{M} \to \Re^+$. To this aim, we exploit the concentration of norms that is dimensionality-dependent. In particular, consider k points $\{\boldsymbol{z}_i\}_{i=1}^k$ drawn from $\mathbf{B}_d$; in [9] the authors show that the pdf associated to the distance r between the hypersphere center $\mathbf{0}_d$ and its nearest neighbor can be written as reported in Eq. (1): since Theorem 1 proves that for $\epsilon \to 0^+$ it is possible to assume uniformly distributed points in every neighborhoods of $\mathcal{M}$ and being ϕ a locally isometric map, the pdf related to the geodetic distances of the embedding space $\Re^D$ converges to the pdf $g(\cdot;\cdot,\cdot)$ in Eq. (1). At this point, as reported in [9], a Maximum Likelihood estimator (ML) could be found for the parameter d of $g(\cdot;\cdot,d)$ (see Eq. (1)), but the resulting estimate may be poor due to the usage of the kNN algorithm, which computes neighborhood estimates of low quality in high dimensions, due to the edge effect [16]. Therefore, in [9] the authors propose to minimize the KL divergence between the pdf computed on the dataset and those calculated on synthetic datasets of known ids; to this aim, they adopt the KL method proposed in [36], a data-driven technique for divergence estimation between multidimensional distributions that is based on nearest neighbor distances.

Though the KL estimation approach proposed in [36] has shown to produce reliable approximations, under our settings the closed-form for the KL divergence between two minimum neighbor distance pdfs can be analytically identified. More specifically, once the parameter k is fixed, we first need to identify the two pdfs by providing an estimate of the parameter d of $g(\cdot;\cdot,d)$; to accomplish this task, we employ the ML estimator [9] described in Section 4. 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 $d \in \{1 \ldots D\}$), 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 KL divergence. Although there exist distributions which do not admit a closed form of the KL divergence, its analytical expression for the minimum neighbor distances may be obtained by integration as follows:

$$\begin{aligned}\overline{KL}_d &= \mathcal{KL}(g(\cdot;k,\hat{d}_{ML}), g(\cdot;k,\check{d}_{d,ML}))\\ &= \int_0^1 g(r;k,\hat{d}_{ML}) \log\left(\frac{g(r;k,\hat{d}_{ML})}{g(r;k,\check{d}_{d,ML})}\right) \mathrm{d}r\\ &= \mathcal{H}_k \frac{\check{d}_{d,ML}}{\hat{d}_{ML}} - 1 - \mathcal{H}_{k-1} - \log\frac{\check{d}_{d,ML}}{\hat{d}_{ML}}\\ &\quad -(k-1)\sum_{i=0}^{k}(-1)^i \binom{k}{i} \Psi\left(1+\frac{i\hat{d}_{ML}}{\check{d}_{d,ML}}\right) \qquad (3)\end{aligned}$$

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

3.2. Concentration of angles

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 $\{\boldsymbol{x}_i\}_{i=1}^{k}$ on a $(d-1)$-dimensional surface S^{d-1} of a hypersphere in $\Re^d$ are subject to the concentration of their values. The common belief that in high dimensions such vectors tend to be orthogonal to each other has found partly justification in the past [37], but only in the last decades an even deeper investigation has allowed a more precise characterization of this fact [38].

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, [39]). More precisely, for $\boldsymbol{x} \in S^{d-1}$, the VMF distribution with parameters $\boldsymbol{\nu}$ and τ has the following density function:

$$q(\boldsymbol{x};\boldsymbol{\nu},\tau) = C_d(\tau)\exp(\tau \boldsymbol{\nu}^T \boldsymbol{x}) \qquad (4)$$

where the unit vector $\boldsymbol{\nu}$ 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 Eq. (4) takes the following form:

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

where I_ν is the modified Bessel function of the first kind with order ν. Due to the normalization factor, this pdf is difficult to be used in theoretical derivations; moreover, in the assumptions of Theorem 1, no information about d may be estimated by the knowledge of the parameters $\boldsymbol{\nu}$ and τ, being $\boldsymbol{\nu}$ uninformative when the hyper-solid angles are uniformly distributed ($\tau = 0$), as in uniformly sampled hyperspheres.

Therefore, to infer the id 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

$$q(\theta;\nu,\tau) = \frac{e^{\tau\cos(\theta-\nu)}}{2\pi I_0(\tau)} \chi_{[-\pi,\pi]}(\theta) \qquad (5)$$

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 [40]. 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 Proposition 1 to show that increasing values of τ are expected for points uniformly drawn from hyperspheres with increasing dimensionality d.

Proposition 1. *Given two independent random unit vectors* $(\boldsymbol{x}_1, \boldsymbol{x}_2)$ *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* $\boldsymbol{x}_1$ *and* $\boldsymbol{x}_2$ *converges asymptotically to the dimensionality d.*

Proof. Consider the following results:

(i) for $d \to \infty$, the random variable $\tilde{\theta} = \sqrt{d}(\theta - \pi/2)$ converges in distribution to a standard normal pdf (see Lemma 3.1 in [38]). In other words, θ converges in distribution to a Gaussian random variable with mean $\pi/2$ and standard deviation $1/\sqrt{d}$;

(ii) 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 $1/\sqrt{\tau}$ [41]. 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 [42], for values of $\tau > 10$ the distribution of the random variable $\theta\sqrt{\tau}$ may be approximated by a standard normal distribution [37]. More accurate approximations [43] sharing the same asymptotic behavior have been introduced by considering further terms in the power series expansion of $\cos(\theta - \nu)$ in Eq. (5).

Item (i) guarantees that θ converges in distribution to a Gaussian random variable with mean $\pi/2$ and standard deviation $1/\sqrt{d}$. Since $\lim_{d\to+\infty} 1/\sqrt{d} = 0$, 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 and (ii) ensure that θ converges in distribution to a Gaussian pdf with mean $\pi/2$ and standard deviation $1/\sqrt{\tau}$. It follows that $\tau \asymp d$, namely $\lim_{d\to+\infty}(\tau/d) = 1$. □

Proposition 1 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. Second, Proposition 1

[1] Note that, due to the kNN bias effect described above, the ML estimates $\check{d}_{d,ML}$ are biased w.r.t. the real value d employed in the sampling process, and a similar bias can be observed also in the estimated $\hat{d}_{ML}$.

allows us to estimate the id of the observed points through the estimation of the concentration parameter τ.

As a further advantage, Proposition 1 suggests that, having to cope with high dimensional manifolds, the id estimate computed by exploiting the concentration of angles is reliable and can be used to enforce the id 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 methodology we propose in Section 4 employs both the ML estimation of the VM parameters ν and τ, and the KL divergence between the VM pdf estimated from the observed dataset and those computed on synthetic data of known ids.

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; more precisely:

$$\hat{\nu} = \text{atan}_2\left(\sum_{i=1}^{N} \sin\theta_i, \sum_{i=1}^{N} \cos\theta_i\right) \tag{6}$$

where $\text{atan}_2(x,y)$ 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 $\eta = I_1(\tau)/I_0(\tau) \equiv A(\tau)$, being η the norm of the sample mean vector defined by Upton [44] as

$$\eta = \sqrt{\left(\frac{1}{N}\sum_{i=1}^{N} \cos\theta_i\right)^2 + \left(\frac{1}{N}\sum_{i=1}^{N} \sin\theta_i\right)^2} \tag{7}$$

Being A a noninvertible function, we rely on the well-known and qualified method proposed in [45], which approximates $A^{-1}(\eta)$ by

$$\hat{\tau} = \tilde{A}^{-1}(\eta) = \begin{cases} 2\eta + \eta^3 + \frac{5\eta^5}{6} & \eta < 0.53 \\ -0.4 + 1.39\eta + \frac{0.43}{1-\eta} & 0.53 \le \eta < 0.85 \\ \frac{1}{\eta^3 - 4\eta^2 + 3\eta} & \eta \ge 0.85 \end{cases} \tag{8}$$

Once an estimate of the VM pdf is obtained, we need to compare it with those computed on synthetic data of known ids. To this aim, a closed-form of the KL between two VM pdfs of parameters ν_1, τ_1, and ν_2, τ_2 is defined in [46] as

$$\overline{KL}_{\nu,\tau} = \mathcal{KL}(q(\cdot;\nu_1,\tau_1), q(\cdot;\nu_2,\tau_2)) = \int_{-\pi}^{\pi} q(\theta;\nu_1,\tau_1)\log\left(\frac{q(\theta;\nu_1,\tau_1)}{q(\theta;\nu_2,\tau_2)}\right)\mathrm{d}\theta$$

$$= \log\frac{I_0(\tau_2)}{I_0(\tau_1)} + \frac{I_1(\tau_1) - I_1(-\tau_1)}{2I_0(\tau_1)}(\tau_1 - \tau_2\cos(\nu_2 - \nu_1)) \tag{9}$$

Note that the introduced framework can be applied for id estimation only if the pdf of angles θ in the embedding space $\Re^D$ converges to the pdf q defined in Eq. (5). This is guaranteed by the local isometry of the map $\phi : \Re^d \rightarrow \Re^D$ embedding a dataset drawn from a manifold $\mathcal{M} = \Re^d$ in a higher dimensional space $\Re^D$ (see Section 3). In fact, the local isometry of ϕ guarantees its conformality with constant dilation factor equal to 1 [47], which intuitively means that a distance preserving map has also the property of preserving angles, as long as the overall area is maintained.

3.3. Combining angle and norm concentration

In the previous sections we described the base theoretic considerations laying foundations for a couple of id estimators exploiting the distinct information conveyed by the concentration of norms and angles. Unfortunately, when considered on their own, both approaches 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 Fig. 1, where we observe a systematic positive bias in the angle-based id estimation which is only loosely counterbalanced by a regular id underestimation based on norms (see Fig. 1(a)). We may read this duality in terms of an opposite behavior of the sensitivity of the angles' compression w.r.t. the dataset id when compared to the norm one, especially in high dimensions (see Fig. 1(b)).

In search of an unbiased id 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 pdfs 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 id estimate we want to compute is

$$\hat{d} = \underset{d \in \{1..D\}}{\text{argmin}} \int_{-\pi}^{\pi}\int_0^1 h_d(r,\theta)\log\left(\frac{h_d(r,\theta)}{\hat{h}(r,\theta)}\right)\mathrm{d}r\,\mathrm{d}\theta$$

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 [48], 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 KL 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 in 3,9, as follows:

$$\overline{KL}_{d,\nu,\tau} = \mathcal{KL}(h_d(r,\theta), \hat{h}(r,\theta)) = \overline{KL}_d + \overline{KL}_{\nu,\tau} \tag{10}$$

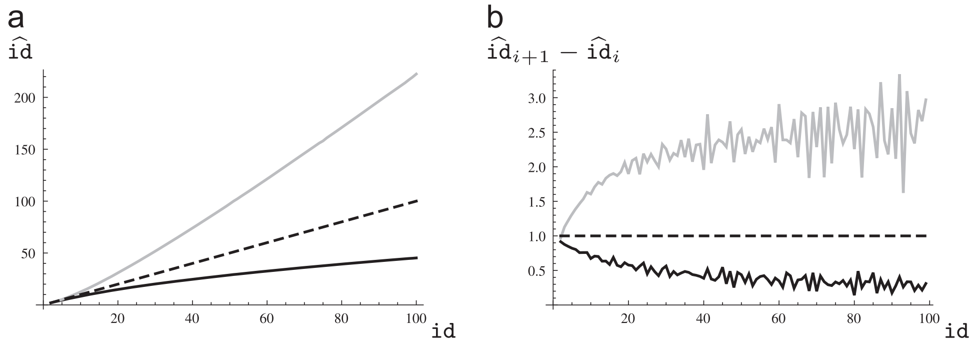

Fig. 1. Comparison between: (a) the id estimates based on angles (gray curve) and norms (black curve) with the exact id values (dashed curve), and (b) the finite differences of the id estimates.

Therefore, an id estimator based on Eq. (10) is obtained by finding the dimensionality d minimizing the KL divergence $\overline{KL}_{d,\nu,\tau}$, hence:

$$\hat{d} = \underset{d \in \{1..D\}}{\arg\min}\, \overline{KL}_{d,\nu,\tau} \tag{11}$$

4. The algorithms

In this section we show how the theoretical considerations presented in Section 3 can be practically exploited to estimate the id of a given dataset combining the informations expressed by angles and minimum neighbor distances. To this aim we collect statistics on the normalized neighborhoods of the kNN graph, and we exploit the information they provide considering them as samples drawn from uniform hyperspheres.

In Section 4.1 a detailed description of the basic algorithm is reported, while Section 4.2 presents its efficient variant, whose time costs are noticeably lower.

4.1. DANCo *algorithm*

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

Under this setting, we extract the information conveyed by the concentration of norms by working on the neighborhood of each point in the dataset; more specifically, for each $\boldsymbol{x}_i \in \boldsymbol{X}_N$, we extract the set of its $k+1$ $(1 \le k \le N-2)$ nearest neighbors $\overline{\boldsymbol{X}}_{k+1} = \overline{\boldsymbol{X}}_{k+1}(\boldsymbol{x}_i) = \{\boldsymbol{x}_j\}_{j=1}^{k+1} \subset \boldsymbol{X}_N$. Calling $\hat{\boldsymbol{x}} = \hat{\boldsymbol{x}}_{k+1}(\boldsymbol{x}_i) \in \overline{\boldsymbol{X}}_{k+1}$ the farthest neighbor of $\boldsymbol{x}_i$, we calculate the distance between $\boldsymbol{x}_i$ and its nearest neighbor in $\overline{\boldsymbol{X}}_{k+1}$, and we normalize it by means of the distance between $\boldsymbol{x}_i$ and $\hat{\boldsymbol{x}}$. More precisely

$$\rho(\boldsymbol{x}_i) = \min_{\boldsymbol{x}_j \in \overline{\boldsymbol{X}}_{k+1}} \frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|}{\|\boldsymbol{x}_i - \hat{\boldsymbol{x}}\|} \tag{12}$$

This equation is used to compute the vector $\hat{\boldsymbol{r}} = \{\hat{r}_i\}_{i=1}^N = \{\rho(\boldsymbol{x}_i)\}_{i=1}^N$ of normalized distances. According to the technique introduced in [9], we compute the ML estimation by numerically solving the optimization problem

$$\hat{d}_{ML} = \underset{1 \le d \le D}{\arg\max}\, ll(d) \tag{13}$$

where

$$ll(d) = N \log\ kd + (d-1) \sum_{\boldsymbol{x}_i \in \boldsymbol{X}_N} \log\ \rho(\boldsymbol{x}_i) + (k-1) \sum_{\boldsymbol{x}_i \in \boldsymbol{X}_N} \log\left(1 - \rho^d(\boldsymbol{x}_i)\right) \tag{14}$$

To this aim we employ the constrained optimization method proposed in [49] 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 $\boldsymbol{x}_i \in \boldsymbol{X}_N$ we find its k nearest neighbors $\overline{\boldsymbol{X}}_k^i$ and we center them by means of a translation to obtain $\hat{\boldsymbol{X}}_k^i = \{\boldsymbol{x}_j - \boldsymbol{x}_i : \forall \boldsymbol{x}_j \in \overline{\boldsymbol{X}}_k^i\}$. Next, we employ the following function:

$$\theta(\boldsymbol{x}_z, \boldsymbol{x}_j) = \arccos \frac{\boldsymbol{x}_z \cdot \boldsymbol{x}_j}{\|\boldsymbol{x}_z\|\ \|\boldsymbol{x}_j\|} \tag{15}$$

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

Since each component of $\hat{\boldsymbol{\theta}}_i$ follows a VMpdfof parameters ν and τ (see Section 3.2), for each set of neighbors we estimate their values by employing the MLapproach described in 6,8, thus obtaining the vectors $\hat{\boldsymbol{\nu}} = \{\hat{\nu}_i\}_{i=1}^N$ and $\hat{\boldsymbol{\tau}} = \{\hat{\tau}_i\}_{i=1}^N$. 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 id. Therefore, for each dimensionality $d \in \{1\ldots D\}$ we uniformly draw a set of N points $\boldsymbol{Y}_{Nd} = \{\boldsymbol{y}_i\}_{i=1}^N$ from the unit d-dimensional hypersphere[2] (named hs^d-sample in the following), and we employ them to compute the vector of normalized distances $\check{\boldsymbol{r}}_d = \{\check{r}_{id}\}_{i=1}^N = \{\rho(\boldsymbol{y}_i)\}_{i=1}^N$ and its MLestimation $\check{d}_{d,ML}$. Next, we calculate the vectors of the VMdistribution parameters $\check{\boldsymbol{\nu}}_d = \{\nu_i\}_{i=1}^N$ and $\check{\boldsymbol{\tau}}_d = \{\tau_i\}_{i=1}^N$ together with their means $\check{\mu}_\nu^d$ and $\check{\mu}_\tau^d$.

Finally, we compose 3,9 as reported in Eq. (10), thus obtaining the following id estimate:

$$\hat{d} = \underset{d \in \{1..D\}}{\arg\min}\, \mathcal{KL}(g(\cdot; k, \hat{d}_{ML}), g(\cdot; k, \check{d}_{d,ML})) + \mathcal{KL}(q(\cdot; \hat{\mu}_\nu, \hat{\mu}_\tau), q(\cdot; \check{\mu}_\nu^d, \check{\mu}_\tau^d)) \tag{16}$$

We call this id estimator DANCo (Dimensionality from Angle and Norm Concentration), and, for the sake of clarity, we report its pseudo-code in the supplementary materials. As shown therein, a further conditional statement has been introduced in order to check whether the id 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 id estimate provided by the aforementioned distances, which can be profitably used in estimating low ids 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 hs^d-sample of growing dimensionality $d \in \{1..D\}$.

4.2. FastDANCo *algorithm*

As shown in Section 5, although outperforming state-of-the-art algorithms, a drawback of DANCo is the long time spent for computing the kNN graph on the hs^d-samples for $d = \{1\ldots D\}$, 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 hs^d-samples.

To this end, we first note that the connection between the dataset in question and the hs^d-samples is extremely loose. In fact, being each 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 (see Section 5.3), after having fixed its value we can generate 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 Fig. 2, 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.

[2] Note that a vector randomly sampled from a d-dimensional hypersphere according to the uniform pdf can be generated by drawing a point $\overline{\boldsymbol{y}}$ from a standard normal distribution $\mathcal{N}(\cdot|\boldsymbol{0}_d, 1)$ and by scaling its norm, see Section 3.29 in [50].

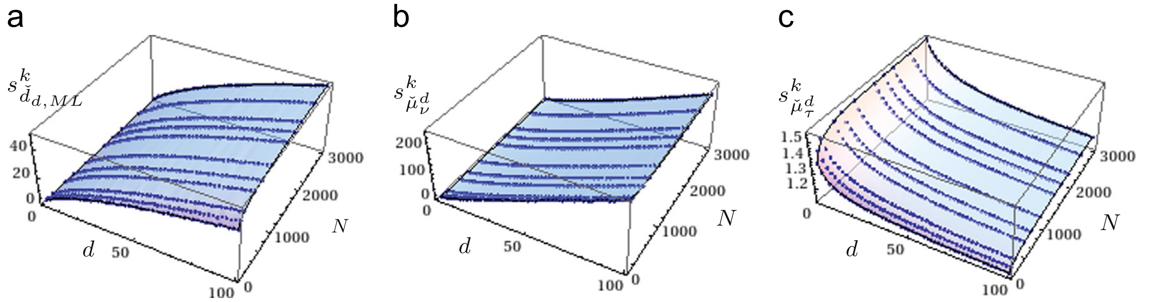

Fig. 2. The cubic smoothing splines fitting the points: (a) $\check{d}_{d,ML}$, (b) $\check{\mu}^d_\nu$, and (c) $\check{\mu}^d_\tau$ w.r.t. the dimensionality d and the sample size N for $k=10$, and averaging over 35 replicas of each $\texttt{hs}^d$-sample.

Table 2
Brief description of the 16 synthetic and 5 real datasets employed in our experiments, where d is the id and D is the embedding space dimension.

Dataset	Name	d	D	Description
Syntethic	$\mathcal{M}_1$	10	11	Uniformly sampled sphere linearly embedded.
	$\mathcal{M}_2$	3	5	Affine space.
	$\mathcal{M}_3$	4	6	Concentrated figure, confusable with a 3d one.
	$\mathcal{M}_4$	4	8	Nonlinear manifold.
	$\mathcal{M}_5$	2	3	2-d Helix
	$\mathcal{M}_6$	6	36	Nonlinear manifold.
	$\mathcal{M}_7$	2	3	Swiss-Roll.
	$\mathcal{M}_8$	20	20	Affine space.
	$\mathcal{M}_{9a}$	10	11	Uniformly sampled hypercube.
	$\mathcal{M}_{9b}$	17	18	Uniformly sampled hypercube.
	$\mathcal{M}_{9c}$	24	25	Uniformly sampled hypercube.
	$\mathcal{M}_{9d}$	70	71	Uniformly sampled hypercube.
	$\mathcal{M}_{10}$	2	3	Möebius band 10-times twisted.
	$\mathcal{M}_{11}$	20	20	Isotropic multivariate Gaussian.
	$\mathcal{M}_{12}$	18	72	Nonlinear manifold.
	$\mathcal{M}_{13}$	24	96	Nonlinear manifold.
Real	$\mathcal{M}_{\mathrm{SiO_2}}$	12	1800	The Crystal Fingerprint space for SiO_2 with 3 atoms.
	$\mathcal{M}_{\texttt{MNIST1}}$	8–11	784	MNIST database (digit 1).
	$\mathcal{M}_{\texttt{SantaFe}}$	9	50	Santa Fe dataset (version $D2$).
	$\mathcal{M}_{\texttt{Isolet}}$	16–22	617	Spoken letter of the alphabet
	$\mathcal{M}_{\texttt{DSVC1}}$	2.26	20	Real time series of a Chua's circuit.

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 $\texttt{hs}^d$-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 $\texttt{hs}^d$-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 smoothing splines. Fig. 2 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 us 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 Supplementary materials). 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 (see Section 5).

5. Algorithm evaluation

In this section we describe the real and synthetic datasets employed in our experiments (see Section 5.1), the experimental settings (see Section 5.2), the comparative evaluation of DANCo and FastDANCo with state-of-the-art id estimators together with the 0 of their robustness w.r.t. their parameter settings (see Section 5.3), the results achieved by significance tests (see Section 5.4) to support the evaluated performances, and the assessment of the robustness of the proposed estimators w.r.t. noise (see Section 5.5).

5.1. Dataset description

To evaluate our algorithms, experiments on 16 synthetic and 5 real datasets (see Table 2) have been performed. In details, to generate 14 synthetic datasets we have employed the tool proposed in [27], and we have further produced the datasets $\mathcal{M}_{12}$ and $\mathcal{M}_{13}$ containing points drawn from nonlinearly embedded manifolds having high id. Precisely, to generate the 2500 points of $\mathcal{M}_{12}$, the applied procedure is the following: we generate 2500 points uniformly drawn in $[0,1]^{18}$; we transform each point by means of $\tan(\boldsymbol{x}^i \cos(\boldsymbol{x}^{18-i+1}))$ where $i=1,\dots,18$; we obtain points in $\Re^{36}$ by appending each transformed $\boldsymbol{x}$ to $\arctan(\boldsymbol{x}^{18-i+1}\sin(\boldsymbol{x}^i))$; we duplicate the coordinates of each point to generate the final points in $\Re^{72}$. The id of the generated $\mathcal{M}_{12}$ is 18, and its points are drawn from a manifold nonlinearly embedded in $\Re^{72}$. To generate $\mathcal{M}_{13}$, which contains 2500 points in $\Re^{96}$, we applied the same procedure on uniformly sampled vectors in $[0,1]^{24}$.

The employed real datasets are the Crystal Fingerprint space for the chemical compound silicon dioxide (SiO_2) with 3 atoms [7], the MNIST database [51], the Santa Fe dataset [52], the Isolet dataset [53], and the DSVC1 time series [6].

The Crystal Fingerprint spaces (or Crystal Fingerspaces) have been recently proposed in crystallography [7] with the aim of representing crystalline structures; these spaces are built starting from the real measured distances between atoms in the crystalline structure. The theoretical id of a Crystal Fingerspace is computed on the $3N_{a+3}$ crystal degrees of freedom, where N_a is the number of atoms in the crystalline unitary cell. In our tests we used the SiO_2 structure with 3 atoms; this dataset has id equal to 12 and contains 4738 points embedded in $\Re^{1800}$.

The MNIST database consists in 70,000 gray-level images of size 28×28 of hand-written digits ($D=784$); in our tests we used the 6742 training points representing the digit 1. Since the id of this dataset is not actually known, we rely on the id estimates proposed in [27,54] for the digit 1, which lie in the range $\{8\ldots11\}$.

The version $D2$ of the Santa Fe dataset is a synthetic time series of 50,000 one dimensional points; it was generated by a simulation of particle motion, and it has nine degrees of freedom. In order to estimate the attractor dimension of this time series, we used the method of delays [55], which generates D-dimensional vectors by collecting D values from the original dataset; by choosing $D=50$ a dataset containing 1000 points in $\Re^{50}$ has been obtained.

The Isolet dataset has been generated as follows: 150 subjects spoke the name of each letter of the alphabet twice, thus producing 52 training examples from each speaker, obtaining a total of 7797 valid samples. The id of this dataset is not actually known, but a study reported in [56] has proposed that the correct id estimate should be in the range $\{16\ldots22\}$.

The DSVC1 is a real data time series composed of 5000 samples measured by means of a hardware realization of the Chua's circuit [57]. We used the method of delays choosing $D=20$, obtaining a dataset containing 250 points in $\Re^{20}$; the id of this dataset is 2.26 [6].

5.2. Experimental settings

To objectively assess our methods,[3] we compared them with the following relevant state-of-the-art id estimators: SPPCA [21], kNNG [32], CD [25], MLE [22], Hein [27], BPCA [18], MiND_{KL} [9], and IDEA [10]. For kNNG, MLE, Hein, BPCA, MiND_{KL}, and IDEA we used the authors' implementation,[4] while for the other algorithms we employed the version provided by the dimensionality reduction toolbox.[5]

To generate the synthetic datasets we adopted the modified generator described in Section 5.1, and we generated for each dataset, 20 instances containing 2500 points. This allowed us to obtain, for each dataset, an unbiased id estimate by averaging over the 20 id estimates. For the same reason, we executed multiple tests also on the real datasets ($\mathcal{M}_{SiO_2}$, $\mathcal{M}_{\text{MNIST1}}$, and $\mathcal{M}_{\text{Isolet}}$) having the greatest cardinality; precisely, for each dataset we extracted 5 random subsets comprising 2500 points each, and we averaged the computed estimates.

Table 3
Parameter settings for the different estimators: k represents the number of neighbors, γ is the edge weighting factor for kNNG, M is the number of Least Square (LS) runs, N is the number of resampling trials per LS iteration, α and π represent the parameters (shape and rate) of the Gamma prior distributions describing 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.

Dataset	Method	Parameters
Synthetic	SPPCA	*None*
	CD	*None*
	Hein	*None*
	MLE	$k_1=6\ k_2=20$
	kNNG_1	$k_1=6, k_2=20, \gamma=1, M=1, N=10$
	kNNG_2	$k_1=6, k_2=20, \gamma=1, M=10, N=1$
	BPCA	$\textit{iters}=500,\ \alpha=(2.0,2.0)\ \pi=(2.0,2.0)\ \mu=(0.0,0.01)$
	MiND_{KL}	$k=10$
	IDEA	$k=10$
	DANCo	$k=10$
	FastDANCo	$k=10$
Real	SPPCA	*None*
	CD	*None*
	Hein	*None*
	MLE	$k_1=3\ k_2=8$
	kNNG_1	$k_1=3, k_2=8, \gamma=1, M=1, N=10$
	kNNG_2	$k_1=3, k_2=8, \gamma=1, M=10, N=1$
	BPCA	$\textit{iters}=2000,\ \alpha=(2.0,2.0)\ \pi=(2.0,2.0)\ \mu=(0.0,0.01)$
	MiND_{KL}	$k=5$
	IDEA	$k=5$
	DANCo	$k=10$
	FastDANCo	$k=10$

In Table 3 the configuration parameters employed in our tests are summarized. To relax the dependency of the kNNG algorithm from the selection of the value of its parameter k, we performed multiple runs with $k_1 \le k \le k_2$ and we averaged the achieved id estimates.

5.3. Experimental results

Table 4 reports the results obtained on the synthetic datasets and shows that the best performing algorithms, which can correctly deal with linear and nonlinear manifolds characterized by low and high id, are DANCo and FastDANCo. In particular, they are the only methods that achieve a good estimation for the datasets $\mathcal{M}_{9d}$, $\mathcal{M}_{12}$, and $\mathcal{M}_{13}$, which have high id and have been drawn from nonlinearly embedded manifolds. On the other hand, the geometric approaches (kNNG, CD, MLE, and Hein) are accurate only on low id manifolds, while the projection techniques (BPCA and SPPCA) are reliable only when linearly embedded manifolds are considered. These results show the underestimation bias generally affecting geometric methods [9,10], and the unreliability of projection methods [22]. Besides, DANCo and FastDANCo outperform also IDEA and MiND_{KL}, which have been purposely developed to deal with datasets having a sufficiently high id (that is id ≥ 10) and being drawn from manifolds nonlinearly embedded in higher dimensional spaces.

In the second last row of Table 4 the Mean Percentage Error (MPE) indicator, proposed in [9] to evaluate the overall performance of a given estimator, is reported. For each algorithm this value is computed as the mean of the percentage errors obtained on each dataset, i.e. $\text{MPE} = 100/\#\mathcal{M}\sum_{\mathcal{M}}|\hat{d}_{\mathcal{M}} - d_{\mathcal{M}}|/d_{\mathcal{M}}$, where $d_{\mathcal{M}}$ is the real id, $\hat{d}_{\mathcal{M}}$ is the estimated one, and $\#\mathcal{M}$ is the number of tested manifolds.

Considering this indicator, DANCo and FastDANCo rank as the best performing estimators.

[3] DANCo and FastDANCo are downloadable from http://www.mathworks.it/matlabcentral/fileexchange/40112-intrinsic-dimensionality-estimation-techniques.

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

[5] http://cseweb.ucsd.edu/-lvdmaaten/dr/download.php.

Table 4
Results achieved on the synthetic datasets. The best approximations are highlighted in boldface. The bottom rows report, for each algorithm, the MPE and the mean execution time (in seconds).

Data	d	SPPCA	BPCA	kNNG$_1$	kNNG$_2$	CD	MLE	Hein	MiND$_{KL}$	IDEA	DANCo	Fast DANCo
$\mathcal{M}_5$	*2*	3.00	**2.00**	1.96	2.06	1.98	1.97	**2.00**	**2.00**	**2.00**	**2.00**	**2.00**
$\mathcal{M}_7$	*2*	3.00	**2.00**	1.97	2.09	1.93	1.96	**2.00**	**2.00**	2.07	**2.00**	**2.00**
$\mathcal{M}_{10}$	*2*	3.00	1.55	1.95	2.03	2.19	2.21	**2.00**	**2.00**	1.98	**2.00**	**2.00**
$\mathcal{M}_2$	*3*	**3.00**	**3.00**	2.95	3.03	2.88	2.88	**3.00**	**3.00**	3.03	**3.00**	**3.00**
$\mathcal{M}_3$	*4*	**4.00**	**4.00**	3.75	3.82	3.23	3.83	**4.00**	4.10	4.01	**4.00**	**4.00**
$\mathcal{M}_4$	*4*	8.00	4.25	4.05	4.76	3.88	3.95	**4.00**	4.05	3.93	**4.00**	**4.00**
$\mathcal{M}_6$	*6*	12.00	12.00	6.46	11.24	5.91	6.39	**5.95**	6.71	6.33	7.00	7.00
$\mathcal{M}_1$	*10*	11.00	5.45	9.16	9.89	9.12	9.10	9.45	10.19	10.41	**10.09**	10.19
$\mathcal{M}_{9a}$	*10*	**10.00**	5.20	8.62	10.21	8.09	8.26	8.90	9.67	9.93	9.86	9.81
$\mathcal{M}_{9b}$	*17*	**17.00**	9.46	13.69	15.38	12.30	12.87	13.85	16.33	16.07	16.62	16.19
$\mathcal{M}_{12}$	*18*	36.00	36.00	14.26	19.8	10.40	12.25	14.10	17.76	16.63	18.76	**18.00**
$\mathcal{M}_8$	*20*	**20.00**	13.55	15.25	10.59	13.75	14.64	15.50	18.24	18.51	19.71	19.00
$\mathcal{M}_{11}$	*20*	**20.00**	13.70	16.40	24.89	11.26	15.82	15.00	18.80	21.20	19.90	**20.00**
$\mathcal{M}_{9c}$	*24*	**24.00**	13.30	17.67	21.42	15.58	16.96	17.95	23.19	23.93	24.28	23.48
$\mathcal{M}_{13}$	*24*	48.00	48.00	17.62	26.87	12.43	14.72	17.76	23.76	18.15	25.76	**24.04**
$\mathcal{M}_{9d}$	*70*	71.00	71.00	39.67	40.31	31.4	36.49	38.69	64.38	46.7	**70.52**	71.00
MPE		35.09	36.03	13.95	17.51	22.05	17.26	12.56	3.32	6.39	2.28	**2.12**
Time		141.21	153.55	84.69	83.91	2.26	**1.19**	1.32	148.00	175.31	66.73	1.61

Table 5
Results achieved on the real datasets by the employed approaches. The bottom rows report, for each algorithm, the MPE and the mean execution time (in seconds). The best results of the last two rows are highlighted in boldface.

Dataset	d	SPPCA	BPCA	kNNG$_1$	kNNG$_2$	CD	MLE	Hein	MiND$_{KL}$	IDEA	DANCo	Fast DANCo
$\mathcal{M}_{\mathrm{DSVC1}}$	*2.26*	4.00	6.00	1.77	1.86	1.92	2.03	3.00	2.50	2.14	2.26	2.00
$\mathcal{M}_{\mathrm{SantaFe}}$	*9*	19.00	18.00	7.28	7.43	4.39	7.16	6.00	7.60	7.26	8.19	8.00
$\mathcal{M}_{\mathrm{MNIST1}}$	*8–11*	9.00	11.00	10.37	9.58	6.96	10.29	8.00	11.00	11.06	9.98	11.00
$\mathcal{M}_{\mathrm{SiO_2}}$	*12*	6.00	3.00	10.24	10.36	1.05	39.28	4.80	17.20	21.20	12.60	12.60
$\mathcal{M}_{\mathrm{Isolet}}$	*16–22*	45.00	19.00	6.50	8.32	3.65	15.78	3.00	20.00	18.77	19.00	19.40
MPE		68.53	68.09	22.96	19.36	49.54	51.86	41.46	13.90	20.37	**2.80**	5.52
Time		2138	1972	1525	1453	15.21	**3.31**	17.73	1610	1840	770	10.95

The last row of Table 4 shows the mean execution time (in seconds) of each algorithm. All the experiments were conducted using a 64-bit Linux 3.2.0-26 operating system running on an Intel Core i7-2670QM CPU, equipped with 8GB of RAM. Note that FastDANCo strongly decreases the time costs of DANCo.

In Table 5 the results obtained on the real datasets are summarized. Being the real data generally affected by the presence of noise, the quality of the estimates computed by the projection methods is strongly reduced, as confirmed by the poor results obtained by BPCA and SPPCA. The geometric approaches we tested are less affected by noise, but most of them are not able to correctly deal with the high dimensionality of $\mathcal{M}_{\mathrm{Isolet}}$.

As can be seen, DANCo and FastDANCo are the best performing estimators, strongly overcoming also the results obtained by those techniques, such as IDEA and MiND$_{KL}$, that exploit a correction-based approach. These results, together with the best average estimation precision achieved by DANCo in terms of MPE,[6] confirm that our methods are promising and valuable tools for id estimation. The last row of Table 5, showing the mean execution time, remarks that FastDANCo is one of the fastest algorithms among those we tested. The speed-up factor is even more noticeable since the employed datasets have very high dimensionality.

Finally, to test the robustness of our algorithms w.r.t. the choice of the parameter k, we employed DANCo to reproduce the experiments proposed for MLE in Fig. 1(a) of [22] and in Fig. 2 of [9], and we averaged the curves obtained in 10 runs. In these tests, the id of datasets composed by points drawn from the standard Gaussian pdf in $\Re^5$ is estimated. We repeated the test for datasets with cardinalities $N \in \{200, 500, 1000, 2000\}$ varying the parameter k in the range $\{5\dots100\}$. As shown in Fig. 3, for all the combinations of the dataset cardinalities and values of k, DANCo obtained id estimates always equal to 5, thus confirming its robustness.

5.4. Tests of significance

To test the significance of differences in performance of the above algorithms, according to the recommendations of [58], we rely on the safe and robust non-parametric Friedman test (FT) followed by a wide family of post-hoc tests to effectively check the over-performance of DANCo w.r.t. the examined competitor algorithms.

However, before analyzing the significance outputted by FT, we may observe the average ranks it computes (see Table 6) and note how DANCo and FastDANCo obtain the highest positions. This superiority is further confirmed by the Friedman statistics, which allows us to decide whether to accept or reject the null hypothesis, H_0, that no significant difference exists in the performances of the compared algorithms. Indeed, the computed statistics decree that H_0 can be rejected with a p-value < 0.0001. Having confirmed the significance of the difference in performance, to check whether DANCo is significantly better than its competitors, we choose it as control method in the series of post-hoc tests available in STATService 2.0 web service [59]. Namely, we perform the set of post-hoc tests as shown in Table 7. The null hypotheses that

[6] When the real id belongs to the range $[d_{min}, d_{max}]$, we calculate the associated MPE term as follows: $\min_{d \in [d_{min}, d_{max}]} |\hat{d}_{\mathcal{M}} - d| / d_{\mathcal{M}}$, where $d_{\mathcal{M}}$ is the mean of the range.

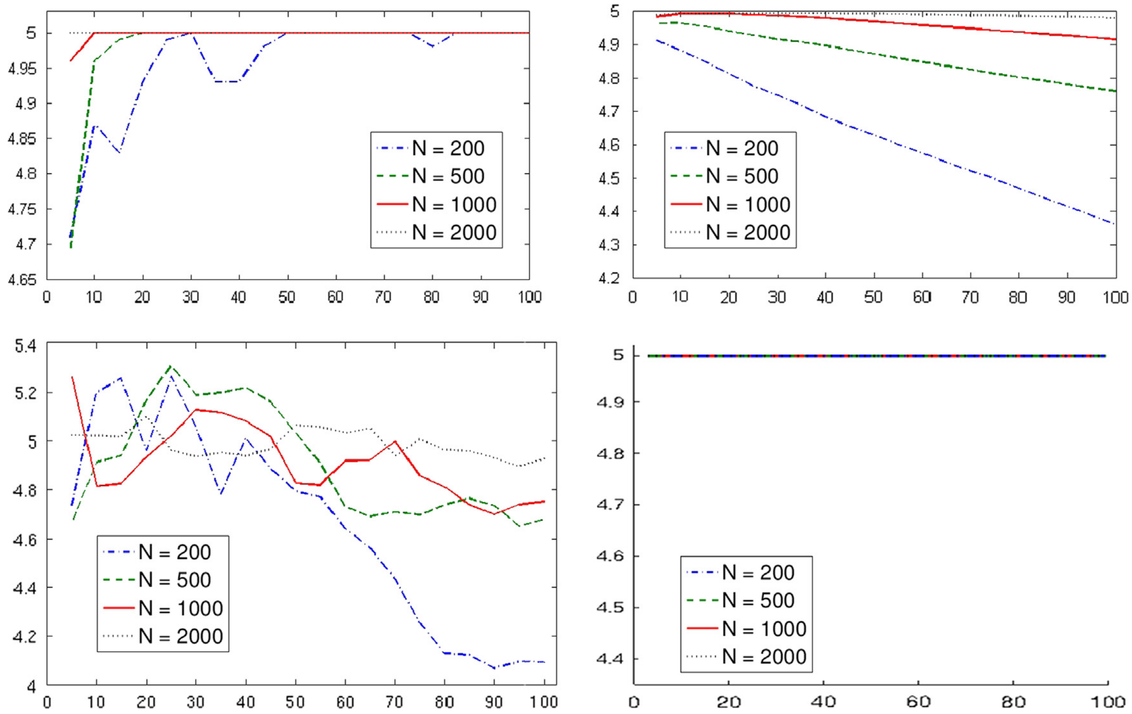

Fig. 3. Behavior of (a) MiND$_{KL}$, (b) MiND$_{MLk}$, (c) IDEA, and (d) DANCo applied to points drawn from a 5-dimensional standard normal distribution; in this test $N \in \{200, 500, 1000, 2000\}$ and $k \in \{5 \dots 100\}$.

Table 6
Average ranks computed by FT.

Algorithm	Average ranks
DANCo	2.538
FastDANCo	3.333
MiND$_{KL}$	4.333
ttIDEA	5.143
Hein	5.905
kNNG$_2$	6.548
SPPCA	6.738
kNNG$_1$	6.952
MLE	7.595
BPCA	8.024
CD	8.905

Table 7
Post-hoc analysis having chosen DANCo as control method. The first row lists the employed significance test method; the first column lists the competitor methods compared to DANCo; and the last row lists the threshold used to reject the null hypothesis H_0.

Test	Bonferroni	Holm	Holland	Rom	Finner	Li
CD	0.0000	0.0050	0.0051	0.0053	0.0051	0.0301
BPCA	0.0000	0.0056	0.0057	0.0058	0.0102	0.0301
MLE	0.0000	0.0063	0.0064	0.0066	0.0153	0.0301
kNNG$_1$	0.0000	0.0071	0.0073	0.0075	0.0203	0.0301
SPPCA	0.0000	0.0083	0.0085	0.0088	0.0253	0.0301
kNNG$_2$	0.0001	0.0100	0.0102	0.0105	0.0303	0.0301
Hein	0.0010	0.0125	0.0127	0.0131	0.0353	0.0301
IDEA	0.0105	0.0167	0.0170	0.0167	0.0402	0.0301
MiND$_{KL}$	0.0771	0.0250	0.0253	0.0250	0.0451	0.0301
FastDANCo	0.4290	0.0500	0.0500	0.0500	0.0500	0.0500
Threshold	*0.0050*	*0.0250*	*0.0253*	*0.0167*	*0.0451*	*0.0301*

DANCo does not perform significantly better than the listed methods is rejected when the corresponding p-value is less than or equal to the thresholds reported in the final row. From Table 7, we not only recognize the superiority of DANCo when compared with the majority of the algorithms, the only exceptions being MiND$_{KL}$ and IDEA for some tests, but we may also note that it is practically indistinguishable from its fast implementation FastDANCo.

To further investigate this aspect, noting that the majority of the examined algorithms succeed in discovering the right dimensionality for small ids, which is a naturally simpler task than that concerning datasets characterized by high ids, we employ the Mann Whitney U Test (Wilcoxon rank-sum test, WT) and the Wilcoxon signed ranked test (WsrT, [58]) to specifically compare the performance of DANCo to those of MiND$_{KL}$ and IDEA on the datasets with $\text{id} \geq 10$. The p-values computed in the comparison between DANCo and MiND$_{KL}$ equal respectively $p_{WT} = 0.0346$ and $p_{WsrT} = 0.0245$, while the comparison between DANCo and IDEA results in $p_{WT} = 0.0482$ and $p_{WsrT} = 0.0009$; these results further confirm that DANCo is significantly superior with a p-value < 0.05. Analogous results are obtained when comparing FastDANCo with the two competitors.

5.5. Tests with noise

In this section we evaluate the robustness of DANCo w.r.t. noise; to this end, we add a zero mean Gaussian noise with stds values σ ranging in $\{0.01, 0.02, 0.05, 0.1\}$ to the synthetic datasets described in Table 2 and we test DANCo by employing the same parameter configurations reported in Table 3. To objectively assess the obtained results we compare them with those achieved by some of the best performing estimators tested in Section 5.3 (Hein, MiND$_{KL}$, and MLE).

From Table 8 it can be noted that our estimators are more sensitive to noise. This might be due to a lower robustness of the statistics computed for id estimation on pairwise angles w.r.t those computed on distances. Nevertheless, taking into account the MPE indicator, the results achieved by DANCo still remain the best ones.

Besides, to check whether the performance degradation observed after noise injection still maintains the same ranking within the compared algorithms, we perform the same hypothesis

Table 8
Experiments on synthetic dataset affected by Gaussian noise with different std σ.

Dataset	d	$\sigma = 0.01$				$\sigma = 0.02$				$\sigma = 0.05$				$\sigma = 0.1$			
		Hein	MiND_{KL}	MLE	DANCo	Hein	MiND_{KL}	MLE	DANCo	Hein	MiND_{KL}	MLE	DANCo	Hein	MiND_{KL}	MLE	DANCo
$\mathcal{M}_1$	10	9.43	9.71	9.15	10.48	9.43	9.52	9.15	10.57	9.71	9.62	9.25	9.05	9.95	10.10	9.50	9.95
$\mathcal{M}_2$	3	3.00	3.00	2.89	3.00	3.00	3.00	2.92	3.00	3.48	3.00	3.07	3.00	4.00	4.00	3.88	4.00
$\mathcal{M}_3$	4	4.00	4.00	3.93	4.00	4.00	4.00	4.09	4.00	4.81	4.33	4.56	4.10	5.00	5.10	5.04	5.00
$\mathcal{M}_4$	4	4.00	4.00	3.96	4.00	4.00	4.05	4.05	4.05	5.00	4.95	4.52	5.00	6.00	5.86	5.54	6.00
$\mathcal{M}_5$	2	2.00	2.00	1.99	2.00	2.00	2.00	2.03	2.00	2.95	2.05	2.19	2.00	3.00	3.00	2.84	3.00
$\mathcal{M}_6$	6	6.00	6.38	6.31	7.00	6.00	6.29	6.37	7.00	6.48	7.05	6.84	7.00	8.90	8.86	8.40	9.14
$\mathcal{M}_7$	2	2.00	2.00	1.98	2.00	2.00	2.00	1.99	2.00	2.00	2.00	2.07	2.00	3.00	2.14	2.21	2.00
$\mathcal{M}_8$	20	15.38	17.71	14.81	19.29	15.43	18.29	14.83	19.48	15.48	17.76	14.83	17.14	15.57	17.43	14.88	17.00
$\mathcal{M}_{9a}$	10	8.95	8.86	8.11	9.86	9.00	8.86	8.52	9.90	9.00	9.24	8.88	9.00	9.48	10.14	9.32	9.76
$\mathcal{M}_{9b}$	17	13.57	14.67	11.82	16.38	13.67	15.14	12.48	16.24	13.95	15.38	13.19	14.81	14.00	16.33	13.8	15.38
$\mathcal{M}_{9c}$	24	17.91	21.86	14.99	23.57	17.91	21.71	15.83	23.67	18.05	22.71	16.95	21.29	18.24	21.43	17.70	21.95
$\mathcal{M}_{9d}$	70	38.14	66.91	29.76	70.43	38.14	64.14	32.10	70.19	38.05	64.19	34.66	68.31	37.62	63.67	36.04	65.52
$\mathcal{M}_{10}$	2	2.00	2.00	2.14	2.00	3.00	2.05	2.22	2.67	3.00	3.00	2.47	3.00	3.00	3.00	2.82	3.00
$\mathcal{M}_{11}$	20	15.00	19.00	16.03	20.00	15.00	18.62	16.00	19.91	15.00	18.43	16.03	19.95	15.00	18.38	16.04	19.86
$\mathcal{M}_{14}$	24	17.81	23.38	15.12	24.57	17.86	22.91	15.51	24.71	19.00	25.00	17.63	23.91	22.52	31.38	23.12	29.67
MPE		12.66	4.19	17.85	2.23	15.65	5.07	17.01	4.50	21.88	10.05	18.11	9.78	28.72	23.10	24.55	22.36

tests as described in Section 5.4. Considering the FT, the null hypothesis H_0 is rejected with a p-value increasing from 0.0001 for $\sigma = 0.01$ and $\sigma = 0.02$, to 0.0025 for $\sigma = 0.05$, and to 0.48 for $\sigma = 0.1$. Moreover, though no significant difference may be observed in the performance of the compared algorithms when high noise levels are added, the post-hoc testing procedure available in STATService 2.0 web service [59], allows us to state that DANCo overperforms both MLE and Hein on the three low-level noise scenarios, while behaving as MiND_{KL} therein.

Besides, following the same steps reported at the end of Section 5.4, we employ the Mann Whitney U Test and the Wilcoxon signed ranked test to compare the two left challengers on those datasets having id ≥ 10. In this case, the computed p-values equal respectively $p_{WT} = 0.0273$ and $p_{WsrT} = 0.0391$ for $\sigma = 0.01$, and $p_{WT} =$ 0.0013 and $p_{WsrT} = 0.0078$ for $\sigma = 0.02$, decreeing once again a significative overperformance of DANCo vs. MiND_{KL} for low-level noises. On the other hand, p-values of $p_{WT} = 0.7911$ and $p_{WsrT} = 1$ for $\sigma = 0.05$, and $p_{WT} = 0.7573$ and $p_{WsrT} = 0.4258$ for $\sigma = 0.1$ witness a similar behavior of the two competitors when noise increases.

6. Conclusions and future works

In this paper we propose a novel estimator, called DANCo, that combines the concentration of angles and norms to estimate the id of a given dataset. The proposed method compares the joint pdf related to angles and norms estimated on the dataset, with those estimated on synthetic datasets of known id. To reduce the time complexity of DANCo we also propose FastDANCo, its faster variant.

We tested our algorithms on both synthetic and real datasets comparing their results with those obtained by employing well-known id estimators. The overall results show that DANCo and FastDANCo are promising and valuable techniques for id estimation since they provide either the best id 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 ids, (ii) manage both linearly and nonlinearly embedded manifolds, and (iii) deal with noisy datasets.

Employing finite set 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 investigations in this direction will be part of our future works. Moreover, since the proposed methods have been designed to identify the id of datasets supposed to lie in a single manifold, our future research works will be also devoted to the extension and/or modification of the proposed methods to cope with underlying multi-manifold structures.

Conflict of interest statement

None declared.

Appendix A. Supplementary material

Supplementary data associated with this article can be found in the online version at http://dx.doi.org/10.1016/j.patcog.2014.02.013.

References

[1] K. Fukunaga, Intrinsic dimensionality extraction, in: P.R. Krishnaiah, L.N. Kanal (Eds.), Classification, Pattern Recognition and Reduction of Dimensionality, 1982.

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

[3] M. Kirby, Geometric Data Analysis: An Empirical Approach to Dimensionality Reduction and the Study of Patterns, John Wiley and Sons, New York, 1998.

[4] V. Vapnik, Statistical Learning Theory, John Wiley and Sons, New York, 1998.

[5] J. Friedman, T. Hastie, R. Tibshirani, The Elements of Statistical Learning – Data Mining, Inference and Prediction, Springer, Berlin, 2009.

[6] F. Camastra, M. Filippone, A comparative evaluation of nonlinear dynamics methods for time series prediction, Neural Comput. Appl. 18 (8) (2009) 1021–1029.

[7] M. Valle, A.R. Oganov, Crystal fingerprint space – a novel paradigm for studying crystal-structure sets, Acta Crystallogr. Sect. A 66 (5) (2010) 507–517.

[8] F. Camastra, A. Vinciarelli, Estimating the intrinsic dimension of data with a fractal-based method, IEEE Trans. Pattern Anal. Mach. Intell. 24 (2002) 1404–1407.

[9] G. Lombardi, A. Rozza, C. Ceruti, E. Casiraghi, P. Campadelli, Minimum neighbor distance estimators of intrinsic dimension, in: Proceedings of ECML-PKDD, vol. 6912, 2011, pp. 374–389.

[10] A. Rozza, G. Lombardi, M. Rosa, E. Casiraghi, P. Campadelli, IDEA: intrinsic dimension estimation algorithm, in: Proceedings of ICIAP, vol. 6978, 2011, pp. 433–442.

[11] A. Rozza, G. Lombardi, C. Ceruti, E. Casiraghi, P. Campadelli, Novel high intrinsic dimensionality estimators, Mach. Learn. 89 (1) (2012) 37–65.

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

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

[14] T. Lin, H. Zha, Riemannian manifold learning, IEEE Trans. Pattern Anal. Mach. Intell. 30 (5) (2008) 796–809.
[15] K. Fukunaga, An algorithm for finding intrinsic dimensionality of data, IEEE Trans. Comput. 20 (1971) 176–183.
[16] P.J. Verveer, R.P.W. Duin, An evaluation of intrinsic dimensionality estimators, IEEE Trans. Pattern Anal. Mach. Intell. 17 (1995) 81–86.
[17] M.E. Tipping, C.M. Bishop, Probabilistic principal component analysis, J. R. Stat. Soc. Ser. B 61 (Part 3) (1997) 611–622.
[18] C.M. Bishop, Bayesian PCA, in: Proceedings of NIPS, vol. 11, 1998, pp. 382–388.
[19] J. Li, D. Tao, Simple exponential family PCA, in: Proceedings of AISTATS, 2010, pp. 453–460.
[20] H. Zou, T. Hastie, R. Tibshirani, Sparse principal component analysis, J. Comput. Gr. Stat. 15 (2004) 265–286.
[21] Y. Guan, J.G. Dy, Sparse probabilistic principal component analysis, J. Mach. Learn. Res.—Proc. Track 5 (2009) 185–192.
[22] E. Levina, P.J. Bickel, Maximum likelihood estimation of intrinsic dimension, Proc. NIPS 171 (2005) 777–784.
[23] K. Pettis, T. Bailey, A. Jain, R. Dubes, An intrinsic dimensionality estimator from near-neighbor information, IEEE Trans. Pattern Anal. Mach. Intell. 1 (1979) 25–37.
[24] P. Mordohai, G. Medioni, Dimensionality estimation, manifold learning and function approximation using tensor voting, J. Mach. Learn. Res. 11 (2010) 411–450.
[25] P. Grassberger, I. Procaccia, Measuring the strangeness of strange attractors, Phys. D: Nonlinear Phenom. 9 (1983) 189–208.
[26] M. Brand, Charting a manifold, Adv. Neural Inf. Process. Syst. 15 (2003) 961–968.
[27] M. Hein, Intrinsic dimensionality estimation of submanifolds in euclidean space, in: Proceedings of ICML, 2005, pp. 289–296.
[28] A.M. Farahmand, C. Szepesvari, J.Y. Audibert, Manifold-adaptive dimension estimation, in: Proceedings of ICML, 2007, pp. 265–272.
[29] B. Kégl, Intrinsic dimension estimation using packing numbers, in: S. Becker, S. Thrun, K. Obermayer (Eds.), Proceedings of NIPSMIT Press, Cambridge, 2002, pp. 681–688.
[30] M. Raginsky, S. Lazebnik, Estimation of intrinsic dimensionality using high-rate vector quantization, in: NIPS, 2005, pp. 1105–1112.
[31] S. Graf, H. Luschgy, Foundations of Quantization for Probability Distributions, Springer, Berlin, 2000.
[32] J.A. Costa, A.O. Hero, Learning intrinsic dimension and entropy of high-dimensional shape spaces, in: Proceedings of EUSIPCO, 2004, pp. 1–22.
[33] J.A. Costa, A.O. Hero, Geodesic entropic graphs for dimension and entropy estimation in manifold learning, IEEE Trans. Signal Process. 52 (8) (2004) 2210–2221.
[34] J. Tenenbaum, V. Silva, J. Langford, A global geometric framework for nonlinear dimensionality reduction, Science 290 (2000) 2319–2323.
[35] J.P. Eckmann, D. Ruelle, Fundamental limitations for estimating dimensions and Lyapunov exponents in dynamical systems, Phys. D: Nonlinear Phenom. 56 (2–3) (1992) 185–187.
[36] Q. Wang, S.R. Kulkarni, S. Verdu, A nearest-neighbor approach to estimating divergence between continuous random vector, in: Proceedings of ISIT, 2006, pp. 242–246.
[37] K.V. Mardia, Statistics of Directional Data, Academic Press, London, 1972.
[38] A. Sodergren, On the distribution of angles between the N shortest vectors in a random lattice, J. Lond. Math. Soc. 84 (3) (2011) 749–764.
[39] K.V. Mardia, P.E. Jupp, Directional Statistics, Wiley Series in Probability and Statistics, John Wiley & Sons, Chichester, 2009.
[40] E. Breitenberger, Analogues of the normal distribution on the circle and the sphere, Biometrika 50 (1-2) (1963) 81–88.
[41] G.J.G. Upton, New approximations to the distribution of certain angular statistics, Biometrika 61 (2) (1974) 369–373.
[42] M. Abramowitz, I.A. Stegun, Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables, ninth edition, Dover, New York, 1964.
[43] G.W. Hill, New approximations to the von Mises distribution, Biometrika 63 (3) (1976) 673–676.
[44] G.J.G. Upton, Approximate confidence intervals for the mean direction of a von Mises distribution, Biometrika 73 (2) (1986) 525–527.
[45] N.I. Fisher, Statistical Analysis of Circular Data, Cambridge University Press, Chichester, 1996.
[46] A.P.N. Vo, S. Oraintara, T.T. Nguyen, Statistical image modeling using von Mises distribution in the complex directional wavelet domain, in: Proceedings of ISCAS 2008, 2008, pp. 2885–2888.
[47] B. O'Neill, Elementary Differential Geometry, Elsevier, Academic Press, San Diego, 2006.
[48] R.D. Lord, The use of the Hankel transform in statistics I. General theory and examples, Biometrika 41 (1/2) (1954) 44–55.
[49] T.F. Coleman, Y. Li, An interior, trust region approach for nonlinear minimization subject to bounds, SIAM J. Optim. 6 (1996) 418–445.
[50] G.S. Fishman, Monte Carlo: Concepts, Algorithms, and Applications, Springer Series in Operations Research, Springer-Verlag, New York, NY, 1996.
[51] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (1998) 2278–2324.
[52] F.J. Pineda, J.C. Sommerer, Estimating generalized dimensions and choosing time delays: a fast algorithm, in: Time Series Prediction. Forecasting the Future and Understanding the Past, 1994, pp. 367–385.
[53] A. Frank, A. Asuncion, UCI Machine Learning Repository, 2010, ⟨http://archive.ics.uci.edu/ml⟩.
[54] J.A. Costa, A.O. Hero, Learning intrinsic dimension and entropy of shapes, in: Statistics and Analysis of Shapes, Birkhauser, 2005, pp. 650–663.
[55] E. Ott, Chaos in Dynamical Systems, Cambridge University Press, Cambridge, 1993.
[56] I. Kivimäki, K. Lagus, I. Nieminen, J. Väyrynen, T. Honkela, Using correlation dimension for analysing text data, in: Proceedings of the ICANN, Springer-Verlag, Berlin, Heidelberg, 2010, pp. 368–373.
[57] L. Chua, M. Komuro, T. Matsumoto, The double scroll, IEEE Trans. Circuits Syst. 32 (1985) 797–818.
[58] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.
[59] J.A. Parejo, J. García, A. Ruiz-Cortés, J.C. Riquelme, Statservice: Herramienta de análisis estadístico como soporte para la investigación con metaheurísticas, in: Actas del VIII Congreso Expañol sobre Metaheurísticas, Algoritmos Evolutivos y Bio-inspirados, 2012.

Claudio Ceruti obtained his master degree cum Laude from the Department of Computer Science of Università degli Studi di Milano in the 2010, where he is a Ph.D. student in applied mathematics. His research interests include machine learning, manifold learning, and dimensionality reduction.

Simone Bassis graduated cum Laude from the Department of Computer Science of Università degli Studi di Milano. In the same department he got the Ph.D. degree in March 2005, and he is an assistant professor. His research interests are in the field of statistic and pattern recognition.

Alessandro Rozza obtained his master degree cum Laude, from the Department of Computer Science of Università degli Studi di Milano/Bicocca in the 2006. He received the Ph.D. degree in Computer Science in March 2011 from the Department of Scienze dell'Informazione, Università degli Studi di Milano. His research interests include machine learning and its applications.

Gabriele Lombardi obtained his master degree cum Laude from the Department of Computer Science of Università degli Studi di Milano in the 2004. In the same University he obtained his Ph.D. in applied mathematics in the 2009. His research interests include machine learning, manifold learning, and computer vision.

Elena Casiraghi graduated cum Laude from the Department of Computer Science of Università degli Studi di Milano in the 2001. In the same department she got the Ph.D. degree in March 2005, and she is an assistant professor. Her research interests are in the medical/biomedical image processing and pattern recognition fields.

Paola Campadelli graduated cum Laude in Biological Sciences from Università degli Studi di Modena in the 1975. In the 1988 she obtained her Ph.D. in Computer Science from the Università degli Studi di Milano. In the period 2007–2012 she has been dean of Faculty of Mathematical, Fisical and Natural Science, Università degli Studi di Milano. Her scientific interests range from machine learning, pattern recognition, and image processing.