Intrinsic Dimension Estimation using Simplex Volumes

Dissertation

zur

Erlangung des Doktorgrades (Dr. rer. nat.)

der

Mathematisch-Naturwissenschaftlichen Fakultät

der

Rheinischen Friedrich-Wilhelms-Universität Bonn

vorgelegt von

Daniel Rainer Wissel

aus

Aschaffenburg

Bonn 2017

Angefertigt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Rheinischen Friedrich-Wilhelms-Universität Bonn

1. Gutachter: Prof. Dr. Michael Griebel

2. Gutachter: Prof. Dr. Jochen Garcke

Tag der Promotion: 24. 11. 2017

Erscheinungsjahr: 2017

Zusammenfassung

In dieser Arbeit stellen wir eine neue Methode zur Schätzung der sogenannten intrinsischen Dimension einer in der Regel endlichen Menge von Punkten vor. Derartige Verfahren sind wichtig etwa zur Durchführung der Dimensionsreduktion eines multivariaten Datensatzes, ein häufig benötigter Verarbeitungsschritt im Data-Mining und maschinellen Lernen.

Die zunehmende Zahl häufig automatisiert generierter, mehrdimensionaler Datensätze ernormer Größe erfordert spezielle Methoden zur Berechnung eines jeweils entsprechenden reduzierten Datensatzes; im Idealfall werden dabei Redundanzen in den ursprünglichen Daten entfernt, aber gleichzeitig bleiben die für den Anwender oder die Weiterverarbeitung entscheidenden Informationen erhalten.

Verfahren der *Dimensionsreduktion* errechnen aus einer gegebenen Punktmenge eine neue Menge derselben Kardinalität, jedoch bestehend aus Punkten niedrigerer Dimension. Die geringere Zieldimension ist dabei zumeist eine unbekannte Größe. Unter gewissen Modellannahmen, zum Beispiel für Punkte auf einer niederdimensionalen Mannigfaltigkeit, wird diese Größe, welche hier der Dimension der Mannigfaltigkeit entspricht, auch als *intrinsische Dimension* der Punktmenge bezeichnet.

Zur Schätzung dieser intrinsischen Dimension existieren diverse Methoden. Viele dieser Verfahren basieren auf der Auswertung lokaler, niederdimensionaler Größen, insbesondere Euklidischer Abstände oder Winkel, welche in Räumen verschiedener Dimension entsprechend unterschiedliche Verteilungen aufweisen und somit Rückschlüsse auf den gesuchten Wert zulassen.

Wir entwickeln einen neuen Ansatz, indem wir die Volumina von Simplexen beliebig hoher Dimension betrachten. Die Eckpunkte eines solchen Simplex werden dabei zufällig aus einer Menge benachbarter Datenpunkte gewählt. Der empirische Mittelwert vieler Volumina wird letztlich dazu genutzt, die zugrunde liegende intrinsische Dimension zu schätzen.

Die Struktur dieser Arbeit lässt sich wie folgt zusammenfassen. Zunächst rekapitulieren und analysieren wir einige Zusammenhänge in hochdimensionalen Räumen, um unser intuitives Verständnis für diese zu schärfen; insbesondere betrachten wir die Entwicklung der Volumina einfacher geometrischer Objekte sowie allgemeine Konzentrationseffekte bestimmter Größen für den Grenzfall einer gegen unendlich strebenden Raumdimension. Anschließend geben wir einen kurzen Einblick in das Thema der Dimensionsreduktion, welche einen Rahmen für das Hauptkapitel bildet.

Im Hauptteil werden zunächst verschiedene Dimensionsbegriffe vorgestellt, des weiteren die wichtigsten Methoden zur Schätzung der intrinsischen Dimension beschrieben und klassifiziert. Eine Auswahl von sechs dieser Methoden wird detaillierter erläutert und dient uns als Benchmark für spätere numerische Untersuchungen. Die konkrete Verfahrensweise unseres eigenen Ansatzes, welchen wir *"Sample Simplex Volume"* nennen, wird sowohl theoretisch motiviert und begründet als auch algorithmisch im Detail geschildert. Eine Kernkomponente ist dabei ein Algorithmus zur schnellen und stabilen Berechnung einer großen Zahl hochdimensionaler Simplex-Volumina. Aufgrund von Laufzeitüberlegungen und Tests mit verrauschten Eingabedaten entwickeln wir eine alternative Variante des ersten Ansatzes und verwenden dementsprechend die Bezeichnungen *SSV1* und *SSV2*.

Wir führen eine Reihe numerischer Experimente sowohl mit zufällig generierten Daten als auch mit frei verfügbaren Datensätzen aus verschiedensten Anwendungen durch. Dabei liegt das Hauptaugenmerk im ersten Fall auf geometrischen Strukturen relativ hoher intrinsischer Dimension, wobei wir auch auf Problembereiche wie *undersampling* (eine im Verhältnis zur Dimension zu geringe Anzahl von Punkten) und verrauschte Daten eingehen. Bei Datensätzen aus konkreten Messungen ist das erwartete Ergebnis der Dimensionsschätzung nicht immer eindeutig. Wir diskutieren die damit verbundenen, teils anwendungsabhängigen Fragestellungen. Unsere eigenen Verfahren erweisen sich in den meisten Fällen als mindestens ebenbürtig zu den zum Vergleich herangezogenen etablierten Techniken.

Danksagungen Zunächst möchte ich Prof. Dr. Michael Griebel dafür danken, dass er diese Arbeit ermöglicht hat, sowie für viele hilfreiche Diskussionen, jegliche fachliche Unterstützung und allgemein die angenehme Arbeitsatmosphäre am Institut für Numerische Simulation. Weiterhin gebührt mein Dank Prof. Dr. Jochen Garcke für wertvolle Anregungen aus dem Bereich des maschinellen Lernens sowie für die freundliche Übernahme des Zweitgutachtens.

Ganz besonders möchte ich an dieser Stelle meine Kollegen des Instituts erwähnen, mit welchen ich zahlreiche mathematische Fragestellungen erörterte, die aber ebenso meinen Arbeitsalltag durch interessante Debatten zu vielfältigen Themen bereicherten. Alexander Rüttgers, Bastian Bohn und Jens Oettershagen sei hier für ihr aufmerksames Korrekturlesen dieser Arbeit gedankt. An die Zeit mit meiner Bürokollegin Jutta Neuen erinnere ich mich gerne zurück, schließlich stand mir bei vielen technischen Problemen Ralph Thesen jederzeit mit seiner Kompetenz und Hilfsbereitschaft zur Seite.

Zu guter Letzt möchte ich meinen Freunden, insbesondere Roman, Yinan, Daniel und Roderich danken, und vor allem auch meinen Eltern, meinem Bruder Johannes, sowie Qiuqiu und Shiyin, die mich in allen Lebenslagen begleitet und unterstützt haben.

Bonn, im August 2017 Daniel Wissel

Contents

1 Introduction **1**

2 High-dimensional Data Analysis and Dimensionality Reduction **7**
2.1 Particularities of High-Dimensional Structures 7
2.1.1 Some considerations on high-dimensional volumes 7
2.1.2 Concentration of measure 11
2.2 Dimensionality Reduction 19
2.2.1 A compact introduction to dimensionality reduction 19
2.2.2 An explanatory example: dimensionality reduction with ISOMAP 21

3 Intrinsic Dimension Estimation **27**
3.1 Concepts of Dimension 27
3.1.1 Lebesgue covering dimension 28
3.1.2 Hausdorff dimension 29
3.1.3 The box-counting dimension 31
3.1.4 The correlation dimension 32
3.1.5 The q-dimension 33
3.1.6 Further notions of dimension 34
3.2 Estimation of the Intrinsic Dimension 34
3.2.1 Classification and characteristics of IDE methods 35
3.2.2 Selected approaches 50
3.3 Simplex Volume Computation 65
3.3.1 Theoretical preliminaries 65
3.3.2 Efficient numerical computation of simplex volumes 69

3.4 Intrinsic Dimension Estimation via Sample Simplex Volumes (SSV) . . . 71
3.4.1 Why simplex volumes? . . . 72
3.4.2 Model and theoretical background . . . 72
3.4.3 Concept of the SSV approach . . . 75
3.4.4 Influence of noise on simplex volumes . . . 82
3.4.5 Algorithmic description of the SSV1 and SSV2 method . . . 84
3.4.6 Complexity analysis of the SSV methods . . . 92
3.5 Numerical Results . . . 94
3.5.1 Challenges of dimension estimation . . . 94
3.5.2 Configuration of the tested methods . . . 97
3.5.3 Technical prerequisites . . . 99
3.5.4 Results from synthetic data . . . 99
3.5.5 Results from real-world data . . . 117
3.5.6 Further numerical evaluations . . . 125
3.5.7 Runtime evaluations . . . 127

4 Application of the SSV Method in Dimensionality Reduction 133

5 Conclusion 139

Bibliography 141

Chapter 1

Introduction

Apart from a small number of prototypes and high-priced, specialized machines, the first personal computers have been developed almost exclusively by American companies in the mid-seventies. Today, about fifty years later, computers have pervaded modern societies triggering changes in various domains, from global to local, e.g. from political, medical, and even ethical issues down to common work routines and habits of individuals. Probably no previous human invention has had a social impact of similar scale within such a short timespan.

The workflow of a computer system can be divided into three basic steps: *input*, *processing*, *output*. While the growth in accumulated processing power of computer devices in their entirety over the last decades has been tremendous, the current quantity and diversity of input devices — and hence the amount of gathered input data — seems at least equally impressive. Due to decreasing hardware manufacturing costs, sensors and other kinds of automatic input devices have recently become omnipresent in modern industrial and consumer products. However, the collection and storage of the increasing amount of resulting data are essentially worthless without the existence of tools to process them in an appropriate way.

This fact underlines the importance of *data mining*. Data mining can be defined as the extraction of "meaningful" information from raw data. Here, the data is already given in a specific context, e.g. in terms of images, documents, categories, or measurements. Data mining methods seek to assist a (usually human) user to detect previously unknown correlations, structures, or similarities in the dataset. The underlying objective can be diverse: categorizing objects, predicting time series, discovering causal relationships, localizing distinctive entities, and many more.

Dimensionality Reduction The first step in many practical data mining scenarios often is the so-called *dimensionality reduction* step. Here, the size of the dataset is reduced by either simply eliminating certain data features or by computing a new, smaller dataset that shares the most relevant attributes with the original one. The two main reasons

for the use of dimensionality reduction methods are the following. First, the data is likely to contain some features that are (almost) irrelevant for the particular objective. Consequently, they might disturb or at least overcomplicate the subsequent data processing step. Second, the sheer size of the original dataset would result in excessive computational costs (time or memory consumption) and thus renders the straightforward application of the respective algorithm impossible.

The task of dimensionality reduction can be considered from two slightly different perspectives: data-driven and model-driven. In the first approach, the given original data is processed "as is", meaning that no a priori assumptions about any underlying structures are made. In contrast, the second approach requires the original data to be generated according to some model, i.e., the data is in fact the outcome of a function usually depending on only a rather small number of generating variables.

In mathematical terms, let the *original data* be given by the finite sequence of points

$$\mathcal{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N), \quad \text{where } \boldsymbol{x}_{i=1,\ldots,N} \in \mathbb{R}^D. \tag{1.1}$$

In the model-driven approach, one assumes that there exists a sequence of *generating variables* of minimal dimension m,

$$\mathcal{Y} = (\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_N) \quad \text{where } \boldsymbol{y}_{i=1,\ldots,N} \in \mathbb{R}^m, \tag{1.2}$$

and a mapping function

$$f : \mathbb{R}^m \to \mathbb{R}^D, \qquad f(\boldsymbol{y}_i) = \boldsymbol{x}_i \ \ \forall\, i = 1, \ldots, N. \tag{1.3}$$

This function might be highly complex, feature many parameters, or even be completely unknown to the user. The task of reconstructing the function f in some appropriate way is the topic of *regression analysis*. While specific knowledge about f can certainly be useful in the process of dimensionality reduction, it is not compulsory. Moreover, in most practical applications, also the dimension m of the generating space $\mathbb{R}^m$ is unknown a priori. Depending on the current perspective, m is referred to as the *number of latent variables* or as the *intrinsic dimension* of the data $\mathcal{X}$.

In the context of machine learning, dimensionality reduction can either emerge as a supervised or an unsupervised task. In the former case, the data $\mathcal{X}$ comes with *labels* $\boldsymbol{L} = (l_1, l_2, \ldots, l_N)$ of some kind which represent the desired output variables of the underlying problem. The pair $(\mathcal{X}, \boldsymbol{L})$ is called the *training data*, and the final objective usually is to infer a function or procedure which maps data points onto labels. This procedure should be able to assign meaningful labels to new data points, but also be compatible with the given training data. In contrast, unsupervised learning is concerned with unlabeled data. Here, the purpose can be preprocessing, compression, denoising, or visualization, compare [LV10].

As an example for supervised learning, consider the collection of human genome data to enhance the understanding of the formation of genetic disorders. Suppose that the

researcher is confronted with two sets of genome data, consisting of genes of affected and non-affected individuals, respectively. The total number of base pairs in each gene set can be as high as several millions. The goal is to identify the (presumably) small number and combination of base pairs relevant for the particular genetic defect. Here, a carefully selected dimensionality reduction method can be used as a first step to eliminate the majority of irrelevant features and thus shrink the original dataset to a size that can be processed further by more sophisticated techniques.

An example for the use of unsupervised dimensionality reduction can be found in hyperspectral imaging. A hyperspectral sensor captures information from multiple bands each featuring a small continuous range across the electromagnetic spectrum. These devices are deployed in such diverse areas as medical examinations, agriculture, mineralogy, astronomy, and food processing. A single measurement can be regarded as a three-dimensional vector with two spatial components and one spectral component. Depending on the analyzed object, certain bands can be highly correlated, and some of them will contain more or less relevant information. Consequently, a dimensionality reduction technique is often applied as a preprocessing step to extract the most useful features.

Other applications include advanced image processing (e.g. recognition of human faces, handwritten digits, traffic signs), forecasting of climatic time series, enhanced investigation of complex chemical and physical simulations or experiments, the training of recommender systems, and many others, compare also [Don00, LV07, GKWZ08].

Intrinsic Dimension Estimation In both dimensionality reduction scenarios presented above, the number of relevant or dominant features is unknown a priori. Yet, numerous reduction methods do not compute a target embedding dimension but rather rely on an external input parameter. Consequently, the estimation of the *intrinsic dimension* of a given dataset is essential for the proper functioning of those methods. As we are primarily interested in the unsupervised case, i.e., our data is unlabeled, let from now on $\mathcal{X}$ denote a *set* of N unique points instead of an ordered sequence as before.

The notion of the intrinsic dimension of some arbitrary set $\mathcal{X} \subset \mathbb{R}^D$ is not a clearly defined concept, neither in mathematics nor in data mining. Nevertheless, precise definitions exist for particular settings. Naturally, for the model introduced above, featuring a mapping function $f : \mathbb{R}^m \to \mathbb{R}^D$ with minimal m, the intrinsic dimension of $\mathcal{X}$ is given by m. In order to enable a rigorous description and a comprehensive analysis of dimensionality reduction as well as intrinsic dimension estimation approaches, one often resorts to the more restrictive model of the data $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ being random point samples of some m-dimensional *manifold* embedded in $\mathbb{R}^D$. In this scenario, while the m-dimensional manifold locally resembles the m-dimensional Euclidean space, the estimation of m is still highly non-trivial due to the impact of manifold curvature, point sampling, and noise.

Common concepts of dimension, such as the Lebesgue covering dimension or the Hausdorff dimension, can not be applied in a straightforward way since they assign the value

of zero to each finite subset of $\mathbb{R}^D$. For this reason, in [GP83], Grassberger and Procaccia introduced the important notion of the *correlation dimension*. The underlying idea of the correlation dimension, as well as of the related *box-counting dimension*, is the following: consider a well-chosen[1] point $\boldsymbol{x}_i$ of the dataset and a relatively small neighborhood, for example the open ball $B_\epsilon(\boldsymbol{x}_i)$ for some small $\epsilon > 0$. Now, supposing that the point set has been sampled from a sufficiently smooth m-dimensional manifold, the number of points $\boldsymbol{x}_j$, $j \neq i$, contained in $B_\epsilon(\boldsymbol{x}_i)$ should be proportional to ϵ^m for increasing ϵ in a suitable interval $\epsilon \in (\epsilon_{\min}, \epsilon_{\max})$.

In fact, while there exist a great amount of diverse techniques to accomplish the task of intrinsic dimension estimation (a contemporary review can be found in [CCCR15]), many of those approaches are based on the fundamental principle of either counting the number of data points in small subdomains or analyzing distance distributions of points in small subdomains. Note that one can basically interpret the counting of data points as a zero-dimensional measurement, whereas the inter-point distance represents a one-dimensional measurement. In theory, most methods based on these low-dimensional quantities allow for the estimation of arbitrarily high intrinsic dimensions m. In practice, many of them have also successfully been employed in applications where $2 \leq m < 10$. However, recent investigations, e.g. in [HA05, BQY13, CCCR15], have revealed a critical problem of numerous estimators: their reliability declines in an overproportional manner for higher values of m. Yet, due to an increasing complexity of modern data mining tasks, this scenario becomes more and more significant.

The question naturally arises whether common low-dimensional measurements, such as distance distributions, could be substituted by high-dimensional equivalents to yield better estimates, especially when it comes to point sets of higher intrinsic dimension. Particular approaches exploring this direction (compare [CCB+14, JSF15]) already achieved some promising results. A reason for the previous lack of such methods might be found in the increased difficulties of analyzing high-dimensional quantities along with numerically evaluating them in an efficient way.

Our new approach is based on the analysis of simplex volumes of arbitrary dimension. The theorem that provides the required theoretical foundations has been established by Miles in [Mil71]. In short, given $s+1$ points drawn at random from the uniform distribution over the n-dimensional unit ball, where $s \leq n$, the theorem specifies the expected value of the (random) volume of the s-simplex spanned by those points. Now, provided that our dataset is sampled from an m-dimensional manifold and under some mild preconditions on the distribution function, it is reasonable to assume that data points in a sufficiently small ϵ-ball are approximately distributed according to the m-dimensional uniform distribution. Consequently, the basic idea is to compare the empirical average volume of multiple simplices, whose vertex points are drawn from local subsets of the data, to the corresponding expected value.

Under the idealized assumption that points from a fixed local subset are perfectly

[1] More precisely, the point should be far away from the boundary of the manifold.

contained in some m-dimensional affine space, the expected volume of the $(m+1)$-dimensional simplices formed as described above equals zero. This fact is exploited in our first, straightforward technique. Here, we increment a *test dimension* $d = 1, 2, \ldots$ and evaluate the associated d-dimensional simplex volumes until d exceeds $\hat{m}$, which shall denote the *estimated* value of the intrinsic dimension.

While this proceeding allows for a very precise estimation of even higher intrinsic dimensions m, it also implicates relatively high computational costs and a distinct sensitivity with respect to noise. For this reason, we develop a second procedure that relies on the analysis of lower-dimensional simplices. It achieves lower runtimes and comes with an increased robustness against disturbances caused by noisy data.

Since, to the best of our knowledge, our approach represents the first of its kind being purely based on simplex volumes, we also describe an efficient numerical technique based on *Cayley-Menger determinants* for the fast evaluation of multiple simplex volumes with vertex points sampled from $\mathbb{R}^D$. The crucial feature of this algorithm is its overall time complexity, which is in fact dominated by the dimension of the considered simplices, but not by the ambient dimension D.

In summary, the s-dimensional simplex is one of the most elementary geometric objects of the s-dimensional space and, furthermore, the simplex volume represents a natural generalization of the Euclidean distance to more than two dimensions. A detailed investigation of its properties for the purpose of dimension estimation appears to be both plausible and valuable. The present work provides a basis for this endeavor.

Contributions This thesis includes the following theoretical and numerical contributions in the context of intrinsic dimension estimation:

- We provide an overview of the vast number of different approaches for intrinsic dimension estimation with a detailed description of selected recent and important variants, which are included in subsequent numerical comparisons.

- One of the core components of our method is the efficient computation of multiple s-dimensional simplex volumes with vertex points in $\mathbb{R}^D$. We present the required theorems from elementary Euclidean geometry and the corresponding algorithm which is both fast and numerically stable. Its overall workload is dominated by the dimension s of the simplices, but not by the ambient dimension D.

- We introduce two variants of our new approach for the purpose of intrinsic dimension estimation called *"Sample Simplex Volume" (SSV)* method. The first variant *(SSV1)* is a straightforward procedure based on arbitrarily high-dimensional simplex volumes, while the second one *(SSV2)* is adapted for better runtime performance and improved handling of noisy datasets. A detailed algorithmic description as well as a complexity analysis are provided for the two versions.

- In an extensive numerical comparison, the SSV methods are evaluated against selected dimension estimators, thus revealing individual advantages and drawbacks. For both synthetical and real-world datasets, our methods feature very competitive estimation results and often outperform most established techniques when it comes to higher intrinsic dimension values.

Outline The remainder of this thesis is organized as follows. In **Chapter 2**, we first discuss certain particularities of high-dimensional objects, which might seem counterintuitive from the naive two- and three-dimensional perspective. Our main focus lies on volumes in Euclidean space and general concentration effects for increasing dimensionality. Subsequently, a condensed introduction to the topic of dimensionality reduction and an explanatory example are given, thus motivating the development of reliable dimension estimators.

Chapter 3 starts with a review of the most important different notions of dimension. Next, a thorough discussion and classification of the multitude of existing approaches for dimension estimation are provided. Selected methods are explained and analyzed in more detail. Subsequently, we introduce our technique for the efficient and stable computation of multiple simplex volumes. The presentation of our SSV methods in section 3.4 includes a theoretical introduction of the underlying concept, the precise algorithmic descriptions, and a runtime complexity analysis. The last section is devoted to numerical experiments involving diverse synthetical and real-world datasets, where our methods are compared with several distinguished dimension estimators.

We revisit the relationship to dimensionality reduction in **Chapter 4** by means of a final illustrating example. **Chapter 5** concludes with a short summary and an outlook on potential future research.

Chapter 2

High-dimensional Data Analysis and Dimensionality Reduction

2.1 Particularities of High-Dimensional Structures

2.1.1 Some considerations on high-dimensional volumes

In some introductions to the field of high-dimensional spaces the reader is confronted with "counter-intuitive" phenomenons or "paradoxical" characteristics of particular high-dimensional structures, see e.g. [Mat02, Ver03]. These formulations are often based on the consideration of familiar objects in two or three dimensions where our human intuition is rather reliable, while the corresponding generalizations of those objects behave more and more unexpectedly for increasing dimensionality. On the other hand, many simple and straightforward relationships exist for one-dimensional geometric structures and we are usually not surprised that these do not hold in the two- or three-dimensional Euclidean space. Keeping in mind that each step from d to $d+1$ dimensions, loosely speaking, "adds" an uncountably infinite number of d-dimensional spaces to the original space, it becomes clear that certain relationships will change rapidly for growing dimension d.

One popular example is the relationship of volumes between the unit hypercube and its inscribed d-ball. It might seem surprising at first that the volume of the corresponding d-ball tends to zero very quickly. However, let us bear in mind that the volume of the surrounding hypercube is fixed to one, while its number of vertices grows exponentially in d, and the inscribed d-ball can not even come close to the vertices since it only touches each facet (or side) of the hypercube in a single point. Therefore, even without looking at the explicit formula for the volume of the d-ball, it seems reasonable that its volume must vanish for higher values of d.

To sharpen our intuition for high-dimensional structures somewhat further, let us consider the analogue relationship as described above, where we now replace the hypercube

polytope	3D object	vertices	facets
hypercube	cube	2^d	$2d$
simplex	tetrahedron	$d+1$	$d+1$
cross-polytope	octahedron	$2d$	2^d

Table 2.1: Properties of selected d-dimensional regular polytopes.

by a regular simplex and a cross-polytope (also called *cocube*), respectively. These three objects are the most simple regular polytopes that can be defined for arbitrary dimension d (compare table 2.1 and figure 2.1). The d-dimensional unit hypercube is given by $[0,1]^d$ and features 2^d vertices as well as $2d$ facets. The regular d-simplex features $d+1$ (affinely independent) vertex points and also $d+1$ facets, while each of its edges has the same length. The d-dimensional (unit) cross-polytope can be defined as the closed unit ball in the ℓ_1-norm on $\mathbb{R}^d$, i.e., $\left\{\boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x}\|_1 \leq 1\right\}$. Its $2d$ vertex points are all permutations of $(\pm 1, 0, 0, \ldots, 0)$ and it has 2^d facets. The constant edge length of this cross-polytope is given by $a = \sqrt{2}$.

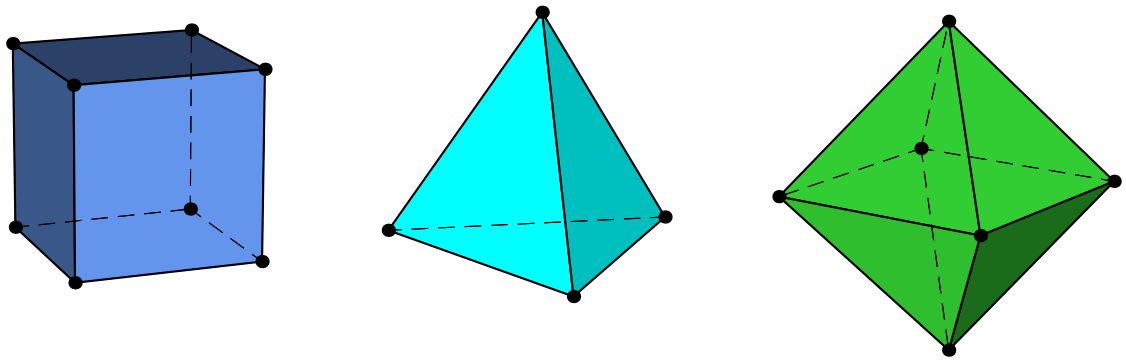

Figure 2.1: Regular 3-dimensional polytopes: cube, regular tetrahedron, and octahedron.

Now, we would like to evaluate the volume of the inscribed d-ball for each of those three polytopes, while their volume is fixed as one. The volume of a d-ball with radius r is given by

$$V_B^{(d)}(r) = \frac{\pi^{d/2} \cdot r^d}{\Gamma\left(d/2+1\right)}. \tag{2.1}$$

Clearly, the radius of the ball inscribed in the unit hypercube is given by $r_H^{(d)} = \frac{1}{2}$. For a regular simplex with edge length a, one can show that its volume and the radius of its

inscribed ball are given by

$$V_S^{(d)}(a) = \frac{\sqrt{d+1}}{d!\sqrt{2^d}}a^d, \qquad r_S^{(d)}(a) = \frac{a}{\sqrt{2d(d+1)}}, \tag{2.2}$$

respectively (see e.g. [EN97]). Letting $V_S^{(d)}(a) = 1$, we can thus evaluate the associated radius as

$$r_S^{(d)} = \left(\frac{d!}{\sqrt{d+1}}\right)^{\frac{1}{d}} \cdot \left(\sqrt{d(d+1)}\right)^{-1}. \tag{2.3}$$

Finally, the volume of a cross-polytope with fixed edge length a and the corresponding radius of its inscribed ball are given by

$$V_C^{(d)}(a) = \frac{\sqrt{2^d}}{d!}a^d, \qquad r_C^{(d)}(a) = \frac{a}{\sqrt{2d}}, \tag{2.4}$$

respectively (see e.g. [TFV15]). Again, letting $V_C^{(d)}(a) = 1$, the associated radius evaluates as

$$r_C^{(d)} = \frac{(d!)^{\frac{1}{d}}}{2\sqrt{d}}. \tag{2.5}$$

Plugging the radii $r_H^{(d)}$, $r_S^{(d)}$, and $r_C^{(d)}$ in formula 2.1 now yields the respective inball volumes in each case. The corresponding plot in figure 2.2 for values $d \in [2, 40]$ reveals some interesting trends. First, we note that for both the hypercube and the simplex, the inball volumes decrease at a rate which is higher than exponential in the dimension d. However the rate of decay is much smaller for the cross-polytope.

Let us offer two intuitive explanations for this phenomenon. From a pure geometric point of view, the inball touches each facet of its surrounding polytope in a single point. Now, a small number of facets does not permit the inball to "grow big" inside the polytope, while on the other hand, a larger number of facets permits the inball to approach the polytope from the inside. In fact, the ball itself can be considered as a regular polytope with infinitely many facets. Consequently, the simplex with its $d+1$ facets contains a relatively small inball, while the inball of the cross-polytope with its 2^d facets is substantially bigger. A second explanation can be based on norms. Obviously, the (solid) cross-polytope, ball, and hypercube can be defined as the point set

$$\mathcal{S}_p(a) := \left\{\boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x}\|_p \leq a\right\} \tag{2.6}$$

for the ℓ_p-norm $\|\cdot\|_p$ with $p = 1, 2, \infty$, respectively. Thus, the q-ball inscribed into a p-ball $\mathcal{S}_p(a)$ is given by

$$\mathcal{S}_q(r), \quad \text{where } r := r_{q,p}(a) = \min_{\left\{\boldsymbol{x}\in\mathbb{R}^d : \|\boldsymbol{x}\|_p = a\right\}} \left(\|\boldsymbol{x}\|_q\right) \tag{2.7}$$

for $1 \leq p \neq q \leq \infty$. In fact, the constrained minimization problem (2.7) can be solved in a straightforward way (at least for $p, q < \infty$) using Lagrange multipliers and the solution evaluates as

$$r_{q,p}^{(*)}(a) = \begin{cases} a \cdot d^{\frac{p-q}{p \cdot q}} & \text{for } q > p, \\ a & \text{for } q \leq p. \end{cases} \tag{2.8}$$

Thus, as expected, we have $\lim_{q \to p} r_{q,p}^{(*)}(a) = a$.

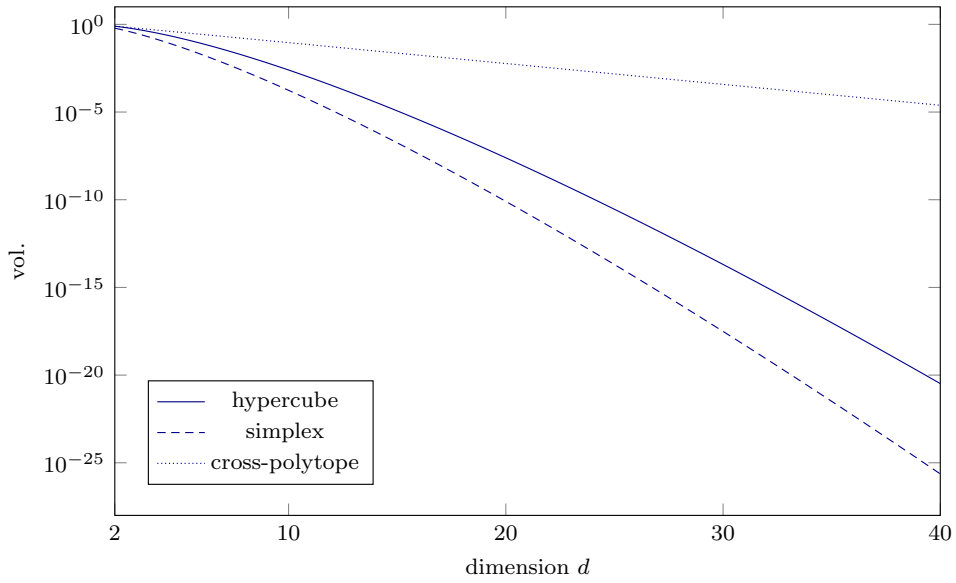

Figure 2.2: Volume of d-ball inscribed in the d-dimensional regular polytope of constant volume 1.

Another often cited example is the volume of the d-ball with radius r relative to the volume of the d-ball with just slightly enlarged radius $r + \epsilon$. Obviously, by virtue of formula (2.1), we have

$$\frac{V_B^{(d)}(r)}{V_B^{(d)}(r+\epsilon)} = \left(\frac{r}{r+\epsilon}\right)^d \to 0 \quad \text{for } d \to \infty. \tag{2.9}$$

For high values of d, the volume thus concentrates in the thin ϵ-shell. Clearly and unsurprisingly, this property generalizes to every d-dimensional geometric object whose volume scales as r^d for some parameter r.

In general, so-called *concentration effects* of various mathematical structures — such as random variables, metrics, or norms — have been studied for a long time. Recently, many of those findings have been rediscovered or refined, and the term "concentration of measure" has been coined in this context.

2.1.2 Concentration of measure

The principle called *concentration of measure* is an important concept in many mathematical fields such as measure theory, probability theory, and combinatorics. First, we aim to give a generic description of this concept. Let us start with a collection of multiple arbitrary objects[1] and a corresponding measure of pairwise (dis-)similarity between them. Now, when considering an increasing number of objects, the measure might emphasize either the *differences* or the *similarities* between the objects to a growing extent. In the first case, the measure still serves as a discriminative function; in the second case, however, it loses its discriminative power and one speaks of *concentration of measure*. It is important to note straightaway that this concentration effect is the result of the *combination* of the particular measure and the considered objects.

In our context of high-dimensional data analysis, it is vital to study the behavior of commonly used metrics for increasing dimension $d \to \infty$. For certain classes of multidimensional objects, different metrics show rather unexpected trends for growing dimensionality. In the following, we provide a compact synopsis of recent advances in this field, which are relevant for our purpose.

First, let us recall the most basic and well-known concentration inequalities of probability theory (compare for example [Geo04]), which in fact represent the foundation of the majority of the subsequent results.

Theorem 2.1 (Markov's inequality)**.** *For a real-valued random variable X with bounded expected value $\mathbb{E}(|X|)$, a constant $\epsilon > 0$, and an increasing function $f : [0, \infty) \to [0, \infty)$ with $f(x) > 0$ for $x > 0$, it holds that*

$$P(|X| \geq \epsilon) \leq \frac{\mathbb{E}(f(|X|))}{f(\epsilon)}. \tag{2.10}$$

Now, letting $f(x) = x^2$ and substituting X by $X - \mathbb{E}(X)$, we immediately get

Theorem 2.2 (Chebyshev's inequality)**.** *Let X be a real-valued random variable with bounded expected value $\mathbb{E}(X)$ and bounded variance $\mathrm{Var}(X) = \mathbb{E}([X - \mathbb{E}(X)]^2)$. Then, for any $\epsilon > 0$, we have*

$$P(|X - \mathbb{E}(X)| \geq \epsilon) \leq \frac{\mathrm{Var}(X)}{\epsilon^2}. \tag{2.11}$$

From those inequalities, one can derive a variant of the weak law of large numbers:

Theorem 2.3. *Let $(X_i)_{i=1,2,\ldots}$ be pairwise uncorrelated random variables with bounded variance and let further $C = \sup_{i \geq 1} (\mathrm{Var}(X_i)) < \infty$. Then, for any $\epsilon > 0$, we have*

$$P\left(\left|\frac{1}{n}\sum_{i=1}^{n}(X_i - \mathbb{E}(X_i))\right| \geq \epsilon\right) \leq \frac{C}{n\epsilon^2} \to 0 \quad \text{for } n \to \infty. \tag{2.12}$$

[1] An *object* in this context shall denote a general (possibly multidimensional) entity, such as a random variable, a vector, or a point of some dataset.

Moreover, for independent and identically distributed (*i.i.d.*) variables X_i, their sample mean $\bar{X}_n := \frac{1}{n}\sum_{i=1}^{n} X_i$ converges to their common expected value $\mu = \mathbb{E}(X_i)$ in probability. A consequence of the law of large numbers is the classical *central limit theorem*, stating that, for *i.i.d.* variables X_i with $\mu = \mathbb{E}(X_i)$, the random variables $\sqrt{n}\left(\bar{X}_n - \mu\right)$ converge in distribution to the Gaussian distribution $\mathcal{N}(0, \sigma^2)$, where $\sigma^2 = \mathrm{Var}(X_i) < \infty$.

Let us now introduce the most important recent findings in the field of concentration of measure, which — to our knowledge — have been presented (among others) in [Dem94, BGRS99, HAK00, AHK01, FWV07, DK09, BM15]. The respective results can actually be categorized according to three different characteristics. The first one is the underlying *probability distribution*, with the most restrictive case of *i.i.d.* random variables. The second is the considered *similarity measure*, ranging from the standard Minkowski norms over ℓ_p-quasi-norms (see below) to general function classes. The third is the actual *quantity* to be analyzed, which can be either a probabilistic (limit) value, e.g. the expectation of a random variable, or some statistical value of a finite set of objects. For the subsequent statements, we require the following definitions.

Definition 2.4 (Quasi-norm)**.** *A quasi-norm $\|\cdot\|$ on a vector space $\mathbb{V}$ over the field $\mathbb{K} = \mathbb{R}$ or $\mathbb{K} = \mathbb{C}$ is a map $\|\cdot\| : \mathbb{V} \to [0, \infty)$ with the properties*

1. *$\|\boldsymbol{v}\| = 0$ if and only if $\boldsymbol{v} = \mathbf{0}$;*
2. *$\|\kappa\boldsymbol{v}\| = |\kappa|\|\boldsymbol{v}\|$ for all $\kappa \in \mathbb{K}$, $\boldsymbol{v} \in \mathbb{V}$;*
3. *$\|\boldsymbol{v} + \boldsymbol{w}\| \leq C\left(\|\boldsymbol{v}\| + \|\boldsymbol{w}\|\right)$ for all $\boldsymbol{v}, \boldsymbol{w} \in \mathbb{V}$ and a given constant $C \geq 1$.*

A quasi-norm differs from a norm as it is required to satisfy only a weaker version of the triangle inequality. We are interested in *ℓ_p-quasi-norms*, where the parameter range is $0 < p < 1$. These are defined exactly like the well-known Minkowski norms of order p (for $1 \leq p \leq \infty$); however, they are no norms since they only fulfill the relaxed triangle inequality with a constant $C > 1$. Note that ℓ_p-quasi-norms are referred to as "fractional distance metric" in [AHK01] and "fractional norm" in [FWV07].

Definition 2.5 (Absolute and relative contrast)**.** *Let $\mathcal{S} = \{\boldsymbol{x}_{i=1,\ldots,N}\} \subset \mathbb{R}^d$ be a finite set of points and let further $\|\cdot\|_*$ denote some norm or quasi-norm defined on $\mathbb{R}^d$. Then we define the* absolute contrast of $\mathcal{S}$ *(with respect to $\|\cdot\|_*$) as*

$$\Omega_{\mathcal{S}}^{(*)} = \Omega_{\mathcal{S}} := \max_i \left\{\|\boldsymbol{x}_i\|_*\right\} - \min_i \left\{\|\boldsymbol{x}_i\|_*\right\}. \tag{2.13}$$

In case that $\mathbf{0} \notin \mathcal{S}$, we define the relative contrast of $\mathcal{S}$ *(with respect to $\|\cdot\|_*$) as*

$$\hat{\Omega}_{\mathcal{S}}^{(*)} = \hat{\Omega}_{\mathcal{S}} := \frac{\max_i \left\{\|\boldsymbol{x}_i\|_*\right\} - \min_i \left\{\|\boldsymbol{x}_i\|_*\right\}}{\min_i \left\{\|\boldsymbol{x}_i\|_*\right\}}. \tag{2.14}$$

Here, each maximum / minimum is taken for all $i = 1, \ldots, N$.[2]

A first result, which has probably drawn more of the data mining community's attention towards concentration phenomenons, has been presented in [Dem94] and studies the Euclidean norm of a random vector of increasing dimensionality, with independent identically distributed components.

Theorem 2.6 (Demartines)**.** *Let $\boldsymbol{x}_d \in \mathbb{R}^d$ be a random vector with i.i.d. components $\boldsymbol{x}_d(k)$, $k = 1, \ldots, d$. Let further their eighth order moment be finite: $\mathbb{E}(\boldsymbol{x}_d(k)^8) < \infty$ for all k. Then, we have:*

$$\mathbb{E}\left(\|\boldsymbol{x}_d\|_2\right) = \sqrt{\alpha d - \beta} + \mathcal{O}\left(1/d\right), \tag{2.15}$$

$$\mathrm{Var}\left(\|\boldsymbol{x}_d\|_2\right) = \beta + \mathcal{O}\left(1/\sqrt{d}\right), \tag{2.16}$$

for $d \to \infty$, with constants α and β that do not depend on the dimension d.

More precisely, the constants α and β depend only on the central moments of order 1 to 4 of a vector component, and thus on the underlying distribution function. The main statement of Theorem 2.6 is, considering only the leading order terms, for $d \to \infty$, the expectation of $\|\boldsymbol{x}_d\|_2$ increases with $\sqrt{d}$, while the variance remains constant. For high values of d, the variance is thus small when compared with the expectation. Demartines infers that high-dimensional random points with *i.i.d.* components appear to be distributed close to a sphere of radius $\mu_d := \mathbb{E}(\|\boldsymbol{x}_d\|_2)$. Furthermore, as the difference of two such points is again a random point with *i.i.d.* components, the Euclidean distance concentrates at the same value μ_d and loses its discriminative power in this scenario.

While the above result is remarkable, it studies a very specific setting. On the contrary, the next theorem provides a conditional statement, involving the relative contrast of some finite point set of growing dimensionality, and requires only very universal assumptions. It has first been presented by Beyer et al. (see [BGRS99]) in a more general form, while the following is a slightly simplified version.

Theorem 2.7. *For each $d = 1, 2, \ldots$, let $\mathcal{F}_d$ be an arbitrary probability distribution on $\mathbb{R}^d$ and, for fixed $N \geq 1$, let further $\mathcal{S}_d = \{\boldsymbol{x}_{d,1}, \ldots, \boldsymbol{x}_{d,N}\} \subset \mathbb{R}^d$ be a (finite) point sample independently drawn from $\mathcal{F}_d$. Finally, let $\|\cdot\|_*$ denote some norm (or quasi-norm) defined on $\mathbb{R}^d$ for each $d \geq 1$ and let $0 < p < \infty$ be a constant. Now, under the condition that*

$$\lim_{d\to\infty} \frac{\mathrm{Var}\left(\|\boldsymbol{x}_{d,i}\|_*^p\right)}{\mathbb{E}\left(\|\boldsymbol{x}_{d,i}\|_*^p\right)^2} = 0, \tag{2.17}$$

we have, for any $\epsilon > 0$,

$$\lim_{d\to\infty} P\left(\hat{\Omega}_{\mathcal{S}_d}^{(*)} \leq \epsilon\right) = 1. \tag{2.18}$$

[2] In the following, when considering the relative contrast of some set $\mathcal{S}$, we implicitly presume $\mathbf{0} \notin \mathcal{S}$ for well-definedness.

In order to judge upon the impact of the statement, it is necessary to identify scenarios fulfilling the prerequisite (2.17). The authors discuss several examples, where, in each case, the norm $\|\cdot\|_*$ is presumed to be the standard Minkowski norm $\|\cdot\|_p$ for $1 \leq p < \infty$. Note here that the parameter p in the above theorem is not related a priori to the used norm $\|\cdot\|_*$.

The first example consists of distribution functions $\mathcal{F}_d$ which are *i.i.d.* in all dimensions and have finite moments up to order $2p$. Second, there are appropriate distributions where each dimension is correlated with all the others and the variance of each additional dimension increases. Another example features distributions, where the variance of each additional dimension tends to zero for $d \to \infty$. Beyer et al. call their Theorem 2.7 an "instability result" and they conclude that the concept of nearest neighbors becomes "meaningless" in high dimensions in many situations that are far more general than *i.i.d.* data.

However, let us discuss two facts showing that the above conclusion must indeed be relativized. First, since the respective result only deals with the limit case $d \to \infty$, the authors provide numerical experiments including the "worst case" scenario of a set $\mathcal{S}_d$ containing $N = 10^6$ uniformly distributed points in dimensions up to $d = 100$; as they do not mention the employed norm, we assume they chose the Euclidean norm. On the one hand, the measured relative contrast decreases (for $d = 1, \ldots, 10$) from roughly $\hat{\Omega}_{\mathcal{S}_1} \approx 10^7$ to only $\hat{\Omega}_{\mathcal{S}_{10}} \approx 8$. On the other hand, even for $d = 100$, they get an empirical value of $\hat{\Omega}_{\mathcal{S}_{100}} \approx 1$, thus relatively far away from the limit value of zero. Note further that the number of $N = 10^6$ points is rather small as a uniform sample of some 100-dimensional space, whereas the expectation of $\hat{\Omega}_{\mathcal{S}_d}$ grows monotonically with increasing N. In summary, given the worst case of uniformly distributed data, while the relative contrast drops rapidly for increasing dimensionality from $d = 1$ to $d = 10$, it is still far from zero for high dimensions such as $d = 100$. Consequently, the question arises whether the problem of "meaningless" distance measures is in fact relevant in practice, where the data is usually far from being uniformly distributed in several hundreds of dimensions.

Second, in [DK09], Durrant and Kabàn provide a "converse theorem" to the above statement and some further enlightening analysis. Their theorem basically states that the converse of the if-then-statement of Theorem 2.7 is also true, under the additional condition that the number N of points is sufficiently large such that

$$\min_{i=1,\ldots,N} \left\{ \|\boldsymbol{x}_{d,i}\|_* \right\} \leq \mathbb{E}\left(\|\boldsymbol{x}_{d,i}\|_* \right) \leq \max_{i=1,\ldots,N} \left\{ \|\boldsymbol{x}_{d,i}\|_* \right\}. \tag{2.19}$$

Further, for a d-dimensional (random) vector $\boldsymbol{x} = (x_1, \ldots, x_d)$ and the standard ℓ_p-norm, the authors reconsider the quotient

$$\frac{\mathrm{Var}\left(\|\boldsymbol{x}\|_p^p \right)}{\mathbb{E}\left(\|\boldsymbol{x}\|_p^p \right)^2} = \frac{\sum_{j=1}^d \sum_{k=1}^d \mathrm{Cov}\left(|x_j|^p, |x_k|^p \right)}{\sum_{j=1}^d \sum_{k=1}^d \mathbb{E}\left(|x_j|^p \right) \cdot \mathbb{E}\left(|x_k|^p \right)}, \tag{2.20}$$

which they refer to as "relative variance" of $\boldsymbol{x}$ with respect to the norm $\|\cdot\|_p$. Nota bene that the notion of "relative variance" has no generally valid definition and is also sometimes used in the field of statistics for quantities differing from the above one.

The authors of [DK09] now argue that all of the examples discussed in [BGRS99] (and satisfying condition (2.17)) feature a sparse correlation structure, where independent components only represent the most trivial case. Hence, the covariances in the numerator of (2.20) are not able to grow at the same (or a higher) rate than the expectations in the denominator with $d \to \infty$ and thus, the limit approaches zero. Durrant and Kabàn further claim that, in contrast to the above examples, real datasets often come with a rich correlation structure and they provide an insightful discussion under which circumstances certain latent variable models do and do not fulfill the critical precondition.

While the previous considerations do not explicitly differentiate between the various norms, it is now particularly interesting to analyze the differing behavior of common (and less common) norms. In this context, Hinneburg et al. (see [HAK00]) study the limit of the *absolute contrast* with respect to the Minkowski norm of order p, in the scenario of random points with *i.i.d.* components.

Theorem 2.8. *Let $\mathcal{F}$ be an arbitrary distribution function on $(0,1)$ and let further $\mathcal{S}_d = \{\boldsymbol{x}_{d,1}, \ldots, \boldsymbol{x}_{d,N}\} \subset \mathbb{R}^d$ be a (finite) set of points, where each coordinate is independently drawn from $\mathcal{F}$. Then, for the Minkowski norm $\|\cdot\|_p$ of order p, we have*

$$C_p \leq \lim_{d\to\infty} \mathbb{E}\left(\frac{\Omega_{\mathcal{S}_d}^{(p)}}{d^{1/p-1/2}}\right) \leq (N-1)\cdot C_p, \tag{2.21}$$

where the constant C_p depends only on p and $\mathcal{F}$.

In our opinion, this theorem is highly interesting since it shows that, in the limit of $d \to \infty$, the absolute contrast $\Omega_{\mathcal{S}_d}^{(p)}$ of a point set with *i.i.d.* components behaves differently in the three cases $p < 2$, $p = 2$, and $p > 2$. More particularly, for commonly used norms, we have in the limit $d \to \infty$:

- $p = 1$: the abs. contrast w.r.t. the *Manhattan norm* scales with $\sqrt{d}$;
- $p = 2$: the abs. contrast w.r.t. the *Euclidean norm* stays within a constant range;
- $p > 2$: the abs. contrast w.r.t. ℓ_p-norms of higher order vanishes.

Consequently, even in the setting of *i.i.d.* components, the widespread ℓ_2-norm does not lose its discriminative power and in fact stands out as the sole ℓ_p-norm for which the expected absolute contrast does not depend on the dimension d of the considered point set.

On the other hand, one could be tempted to presume that an absolute contrast growing with the underlying dimension d, as for the Manhattan norm, should generally be beneficial for the analysis of high-dimensional data, due to a better discrimination of

neighboring points. This supposition also motivated HINNEBURG et al. in [AHK01] to consider the ℓ_p-quasi-norms for $p \in (0,1)$, which they refer to as "fractional distance metrics". From now on, we simply speak of *ℓ_p-(quasi-)norms* for $0 < p < \infty$, when in fact subsuming the quasi-norms ($p < 1$) and norms ($p \geq 1$). For the (most restrictive) scenario of *uniformly distributed* points, HINNEBURG et al. provide the following bounds:

Theorem 2.9. *Let $\mathcal{S}_d = \{\boldsymbol{x}_{d,1}, \ldots, \boldsymbol{x}_{d,N}\} \subset \mathbb{R}^d$ be a set of uniformly distributed points. Let further $\|\cdot\|_p$ denote the associated be ℓ_p-(quasi-)norm for $0 < p < \infty$. Then, the absolute and the relative contrast of $\mathcal{S}_d$, with respect to the (quasi-)norm $\|\cdot\|_p$, behave as follows:*

$$\frac{C_1}{(p+1)^{1/p} \cdot \sqrt{2p+1}} \leq \lim_{d\to\infty} \mathbb{E}\left(\frac{\Omega_{\mathcal{S}_d}^{(p)}}{d^{1/p-1/2}}\right) \leq \frac{(N-1)\cdot C_1}{(p+1)^{1/p} \cdot \sqrt{2p+1}}; \tag{2.22}$$

$$\frac{C_2}{\sqrt{2p+1}} \leq \lim_{d\to\infty} \mathbb{E}\left(\hat{\Omega}_{\mathcal{S}_d}^{(p)} \cdot \sqrt{d}\right) \leq \frac{(N-1)\cdot C_2}{\sqrt{2p+1}}, \tag{2.23}$$

where C_1 and C_2 are some constants.

This theorem shows that, for $d \to \infty$, a quasi-norm with small $p < 1$ potentially allows a larger range for the expected (both absolute and relative) contrast of N uniformly distributed points.

For that reason, the authors of [AHK01] advocate the use of quasi-norms in certain high-dimensional data mining applications and perform the following experiment. They select a small number of datasets from the *UCI Machine Learning Repository* [Lic17], which all stem from classification problems. Now for each point $\boldsymbol{x}_i$ of a given set $\{\boldsymbol{x}_{i=1,\ldots,N}\}$, they count the number $n_i(p)$ of nearest neighbor points — with respect to the ℓ_p-(quasi-)norm — that feature the same class label as $\boldsymbol{x}_i$. The corresponding sum $\bar{n}(p) := \sum_i n_i(p)$ is then suggested as some kind of quality measure of the respective ℓ_p-(quasi-)norm. For all tested values of $p \in \{0.1, 0.5, 1, 2, 4, 10, \infty\}$ and each tested dataset, the empirical results fulfill $\bar{n}(p_1) > \bar{n}(p_2)$ for $p_1 < p_2$ (except for two cases). The authors therefore conclude that smaller values of p generally seem to be better suited to measure dissimilarities of high-dimensional data. They argue that norms with higher values of p tend to overemphasize the influence of single dimensions (with high variance), while lower values of p rather extenuate the influence of those "outlier dimensions".

While this reasoning is essentially plausible, further practical studies have come to more deliberate conclusions. In [DAD04], the authors subsume that their results on K-means class recovery show no superiority of the tested ℓ_p-quasi-norm (for $p = 0.3$) when compared to the standard ℓ_2-norm. Moreover, they replicated the experiments of [AHK01] described above, however with d-dimensional data that has been standardized in the sense that each component has been shifted and scaled as

$$\bar{x}_j := \left(x_j - x_j^{(\min)}\right) / \left(x_j^{(\max)} - x_j^{(\min)}\right),$$

where $x_j^{(\min)}$ and $x_j^{(\max)}$ denote the respective minimum and maximum of the jth component. For this type of standardized data, again, the considered quasi-norms and the Euclidean norm feature comparable outcomes. In [HR05], the authors investigate the practical advantages of quasi-norms in the field of image retrieval and conclude that, while they could not derive a scheme to find the optimal value of p for a given dataset, the ℓ_p-quasi-norms for values $p \in [0.25, 0.75]$ achieved the best results in most scenarios. More precisely, quasi-norms generally seem to be favorable when it comes to sparse vectors, while for dense vectors, the differences vanish. A similar insight is presented in [FWV05], where the relationship between noise characteristics of datasets and the suitability of different (quasi-)norms is examined. According to the authors, the Euclidean norm is superior for data tainted with white (i.e., Gaussian) noise, while quasi-norms are preferable for *colored noise*, i.e., noise that affects not all, but only some components to a high degree. An example is the so-called "salt and pepper noise" of images, consisting of sparsely scattered black or white erroneous pixels.

While Theorem 2.9 only holds for uniform distributions, in [FWV07] and [BM15], further concentration results of ℓ_p-(quasi-)norms are given for the slightly more general case of points with *i.i.d.* components. However, these statements completely rely on the assumption of independent components, which usually does not apply for real-world data. François et al. conclude that — depending on the underlying distribution functions — "fractional norms" are not always less concentrated than norms of higher order. Moreover, they discuss some approaches to determine the (in an empirical sense) optimal value p of the (quasi-)norm depending on certain dataset characteristics.

Recently, a growing number of publications have studied the interplay of different ℓ_p-(quasi-)norms and real-world data in more detail, compare e.g. [FCX12, Kab12]. In [HC09], an interesting ansatz to circumvent concentration phenomenons is propagated. The authors introduce a so-called "shrinkage-divergence proximity" function. For d-dimensional points $\boldsymbol{x} = (x_1, \ldots, x_d)$, $\boldsymbol{y} = (y_1, \ldots, y_d) \in \mathbb{R}^d$, and some given metric $\delta : \mathbb{R} \times \mathbb{R} \to [0, \infty)$, they define the similarity measure

$$s(\boldsymbol{x}, \boldsymbol{y}) = \sum_{j=1}^{d} w_j \cdot f_{a_j, b_j}(\delta(x_j, y_j)),$$

where the functions $f_{a,b} : [0, \infty) \to [0, \infty)$ are given by

$$f_{a,b}(z) = \begin{cases} 0, & \text{if } 0 \leq z < a \\ z, & \text{if } a \leq z < b \\ e^z, & \text{otherwise,} \end{cases}$$

and w_j, a_j, b_j are parameters that can be adapted to the dataset. Consequently, large dissimilarities in the components are overemphasized, while small ones are ignored. The authors do not offer any advanced theoretical analysis of the properties of their proximity

function, but they present various experimental results based on a k-NN classifier. On the one hand, the proposed similarity function outperforms both the ℓ_2-norm as well as the $\ell_{1/2}$-quasi-norm in most of their experiments; on the other hand, its performance is highly dependent on the numerous parameters. Nevertheless, the underlying idea of adapting similarity measures to (high-dimensional) datasets demonstrates an interesting direction for further research.

Let us now give a compact summary of the diverse facets of the concentration of measure phenomenon discussed above. First of all, we would like to emphasize here again that concentration, i.e., the loss of discriminative power, is always a consequence of the *interplay* of a particular measure and a particular object (e.g. random vector, dataset). Many conclusions of norms becoming concentrated in high dimensions rely on very restrictive assumptions, often random vectors with *i.i.d.* components. The situation is much more complex for arbitrarily distributed points, and thus, for real-world data mining applications. A similar reasoning applies to the ℓ_p-(quasi-)norms. While there is good evidence that Minkowski norms of higher order $p > 2$ might be less useful in high-dimensional spaces, many different scenarios have been described where either the commonly-used ℓ_1- and ℓ_2-norms or certain quasi-norms with $p < 1$ turn out as favorable. However, the topic of identifying the (in some sense) optimal value of p with respect to a given dataset — or even the optimal dissimilarity measure in a more general sense — still leaves a lot of space for future investigations.

Another aspect that has been observed for example in [FWV07] and [DK09] is the fact that concentration effects of norms naturally scale with the *intrinsic* dimension of the data rather than the ambient dimension. The different interpretations of this term will be discussed in Chapter 3. For the sake of clarity, consider the trivial case of points sampled from an m-dimensional affine subspace embedded in $\mathbb{R}^D$ with $m \ll D$. Clearly, the distances between those D-dimensional points resemble distances between m-dimensional points. Thus, in this scenario, the high ambient dimension D does not affect the norm concentration at all.

In data mining, when the underlying structure of the dataset is unknown, the estimation of the intrinsic dimension and subsequent reduction of the ambient dimension are often the first two steps before the dataset is processed further on. Without any a priori knowledge, as we have seen above, there is no reason to favor the use of a particular (quasi-)norm. As a consequence, most general-purpose methods both for intrinsic dimension estimation (IDE) as well as for dimensionality reduction (DR) are based on the standard ℓ_2- or (sometimes) the ℓ_1-norm. Nevertheless, it should be kept in mind that quasi-norms can be advantageous in specific settings.

The main topic of this thesis is the theoretical and practical introduction of a new approach for the efficient and reliable estimation of the intrinsic dimension of a given dataset. However, the task of IDE is highly interrelated with dimensionality reduction, which is why we choose to embed our main discussion into an introductory example in order to illustrate the interplay between the two concepts.

2.2 Dimensionality Reduction

2.2.1 A compact introduction to dimensionality reduction

Since a large number of different approaches and techniques have been attributed to the field of dimensionality reduction (DR), it surely is helpful to outline certain characteristics that allow for a first classification and categorization of those methods. For this purpose, we make use of the carefully elaborated proposal presented in [LV07] by Lee and Verleysen.

To establish the general setting, let us start with a given dataset of N objects with D components or features each, where D is large. These features might be of different type, e.g. numeric, categorical, textual or other. The dataset shall now be used or analyzed or processed in a particular context. In this context, some of the available features might be more or less relevant, others might be completely irrelevant. The separation of relevant and irrelevant features, also sometimes referred to as *feature selection*, is often accomplished in a supervised manner and is not studied any further here.

Given only the relevant features, there might still be relationships between them: if one particular feature changes in a certain way, another feature changes accordingly. From the stochastic viewpoint, these two features are correlated or at least not independent. Consequently, the general assumption is that there exists a smaller number $m < D$ of so-called *latent variables* and a rule set mapping those variables onto the original features, thus recovering the original dataset. Here, the number m of latent variables is usually unknown, as well as the respective rule set, which might range from a simple linear projection to some highly complex function. Besides, the recovery of the given data can also be of approximate nature.

Dimensionality reduction methods basically seek to construct a new, low-dimensional dataset that allows to recover the relevant structure of the original (high-dimensional) dataset in the best possible manner. These methods can now be classified according to the following three specific tasks or functionalities.

1. Estimation of the number of latent variables.
2. Mapping the data points into low-dimensional space to reduce their dimensionality.
3. Mapping the data points into low-dimensional space to recover the latent variables.

The estimation of the number of latent variables is a crucial part of the whole process. Some DR methods feature their own scheme to derive a proper estimate, others at least come with some technique that allows to narrow down the choice, while the majority rely on an external input parameter. As explained above, the notion of "latent variables" assumes the existence of a corresponding function projecting those variables into the original data space. However, in many practical scenarios, there is no need to study this function which in fact is often likely to be of extremely high complexity. In this case, one rather speaks of the *intrinsic dimension* of the dataset instead of the number of latent

variables. Here, the precise representation of the latent variables and the mapping are no longer relevant, but only the pure (topological) dimension of the reduced data. The estimation of this intrinsic dimension is the main topic of Chapter 3.

The second task describes the case where the goal of dimensionality reduction is to construct a low-dimensional representation with minimal target dimension, but without involvement of the latent variables. Here, the underlying practical purpose is often data compression, visualization, or preprocessing. In contrast, the third task includes the recovery of latent variables. Sometimes, the process is split into dimensionality reduction and latent variable separation. Concerning the latent variables, frequently, certain statistical properties, e.g. the independence of components, are requested.

A further categorization of DR techniques can be accomplished by considering the underlying model the data are presumed to follow, the precise algorithmic procedure or implementation, and the optimization criterion.

Concerning the model, one distinguishes e.g. *linear* from *nonlinear*, as well as *continuous* from *discrete* approaches. We stick to the naming convention of "linear" versus "nonlinear" dimensionality reduction methods, when actually referring to the underlying linear / nonlinear model. The most well-known and widespread linear methods include *principal component analysis* (PCA) and (classical) *multidimensional scaling* (MDS), as well as *linear discriminant analysis* (LDA), all of those featuring many different variants under various names. Since PCA and MDS also include a technique to derive an estimate of the intrinsic dimension, they are discussed in more detail in subsection 3.2.1. A comprehensive survey on linear DR methods can be found in [CG15].

Linear methods come with obvious limitations, however are still often used in practice, either as a first pre-processing step to reduce the dimensionality of very high-dimensional data, or just because of their simplicity. Nevertheless, to overcome these immanent limitations, a growing number of nonlinear approaches have been coined in the last decades. These approaches can be loosely categorized into *distance preserving* and *topology preserving* methods.

The first group can be subdivided further according to the used distance type, such as e.g. the Euclidean or the geodesic distance. The use of the Euclidean metric is often linked to a well-defined optimization criterion and thus allows for a precise theoretical analysis of the method's characteristics. Geodesic distances, usually approximated by graph distances when dealing with a finite number of sample points, seek to reproduce the geometric shape of the underlying manifold and may outclass the Euclidean model considerably in appropriate scenarios.

While distance preserving methods focus upon the relationship of pairs of points, the model of topology preserving methods is based on the global topology of the manifold. In the precise implementation, the discretization step usually involves either a lattice or a graph structure. Even though topology preserving approaches are often able to uncover complex underlying structures due to their flexible model assumptions, most of them are restricted in practice to rather (both extrinsically and intrinsically) low-dimensional problems, because of their algorithmic and parametric complexity.

Plenty of literature on both linear and nonlinear dimensionality reduction methods exists. Two nice compact reviews can e.g. be found in [vdMPvdH09] and [SVP14], while for a more comprehensive overview, we again refer to [LV07].

2.2.2 An explanatory example: dimensionality reduction with ISOMAP

The ISOMAP method has been introduced by Tenenbaum, de Silva and Langford in [TdSL00] and belongs to the group of nonlinear dimensionality reduction methods preserving geodesic distances. More precisely, ISOMAP is based on the so-called *(classical) multidimensional scaling* (compare [Tor52]), which is per se a linear method. However, replacing the Euclidean distances of the classical variant with geodesic distances makes ISOMAP a nonlinear approach. Miscellaneous extensions of the method have been proposed e.g. in [dST03, SGM04, CC07].

The term "multidimensional scaling" (MDS) in fact refers to a whole category of different procedures which are discussed in further detail in subsection 3.2.1. At this point, let us just give a short description of classical MDS.

For a given dataset $\mathcal{X} = \{\boldsymbol{x}_{i=1,\dots,N}\} \subset \mathbb{R}^D$, consider the corresponding centered points

$$\bar{\boldsymbol{x}}_i := \boldsymbol{x}_i - \frac{1}{N}\sum_{j=1}^{N} \boldsymbol{x}_j,$$

and the $(N \times D)$-matrix X made up of those $\bar{\boldsymbol{x}}_i$. The Gram matrix $B = X \cdot X^T$ is positive-semidefinite and symmetric and the crucial component is now its eigenvalue decomposition

$$B = S\Lambda S^T, \quad \text{with } \Lambda = \operatorname{diag}(\lambda_1, \dots, \lambda_N),$$

where at most D eigenvalues λ_i are positive and the remaining ones are zero. The dimension reduction step consists of selecting a number m (with $m < D$) of components to be kept and the reduced data is then given by

$$\hat{X} = S_m \Lambda_m^{1/2},$$

where the $(m \times m)$-matrix Λ_m contains the largest eigenvalues on its diagonal and the $(N \times m)$-matrix S_m holds the associated eigenvectors. It must further be noted that the Gram matrix B can be rewritten as

$$B = -\frac{1}{2} JQJ,$$

where J is some centering matrix and $Q = [q_{ij}]$ with $q_{ij} = \|\bar{\boldsymbol{x}}_i - \bar{\boldsymbol{x}}_j\|^2$ is the squared distance matrix. Thus, given the matrix Q, the MDS computations can in fact be accomplished without the explicit knowledge of the precise data point coordinates.

In the ISOMAP algorithm, the Euclidean distances are now replaced by graph distances, i.e., approximated geodesic distances. For this purpose, a weighted nearest neighbor (NN) graph G is constructed for the input points, either a k-NN or an ϵ-ball graph. For the first variant, a point $\boldsymbol{x}_j$ is connected to $\boldsymbol{x}_i$ if it is among its k nearest neighbors, for the second variant, if $\|\boldsymbol{x}_i - \boldsymbol{x}_j\| < \epsilon$. The corresponding edge weight is set to $\|\boldsymbol{x}_i - \boldsymbol{x}_j\|$ in each case. The final graph distance between two points is now given by the sum of all edge weights on the shortest path in G between the points.

Swiss role and heated swiss role

Let us now consider two different scenarios to shed some light upon the abilities and shortcomings of the ISOMAP dimensionality reduction algorithm. First, we revert to the so-called "swiss role" dataset that has indeed been analyzed in [Ten98] to demonstrate the advantages of ISOMAP and has quickly become a popular benchmark case study for both dimension reduction and dimension estimation approaches. The dataset, a two-dimensional manifold embedded in three-dimensional space, resembles a coiled up sheet and its name is derived from a certain type of sponge cake. To our knowledge, there exists no standardized parameterization of this dataset. One way to generate the data points is to use the function $f_{\text{sw}} : \mathbb{R}^2 \to \mathbb{R}^3$,

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \mapsto \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} \sqrt{y_1} \cdot \cos\left(4\pi\sqrt{y_1}\right) \\ \sqrt{y_1} \cdot \sin\left(4\pi\sqrt{y_1}\right) \\ y_2 \end{bmatrix}, \tag{2.24}$$

where $[y_1, y_2]^T$ is uniformly distributed in $[0, 1]^2$. A more complex variant, the so-called "heated swiss role", has been proposed in [LV07], and can be described via the function $f_{\text{swh}} : \mathbb{R}^2 \to \mathbb{R}^3$,

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \mapsto \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} (2y_2^2 - 2y_2 + 1) \cdot \sqrt{y_1} \cdot \cos\left(4\pi\sqrt{y_1}\right) \\ (2y_2^2 - 2y_2 + 1) \cdot \sqrt{y_1} \cdot \sin\left(4\pi\sqrt{y_1}\right) \\ y_2 \end{bmatrix}, \tag{2.25}$$

where, again, $[y_1, y_2]^T$ is uniformly distributed in $[0, 1]^2$. Here, the y_2 variable is used to yield a height-dependent parabolic bending of the structure.

While at first sight, both manifolds resemble each other quite closely, the swiss role represents a so-called *developable* manifold (in the terminology of differential geometry), while the heated version is *non-developable*. Basically, an m-dimensional manifold is called *developable* if there exists an isometry between geodesic distances on the manifold and Euclidean distances within a convex subset of $\mathbb{R}^m$. Intuitively, if the manifold can be "unfolded" without any distortions, it is developable. This can be done for the swiss role, but not for the heated version. Another simple example of a non-developable manifold is of course the m-sphere.

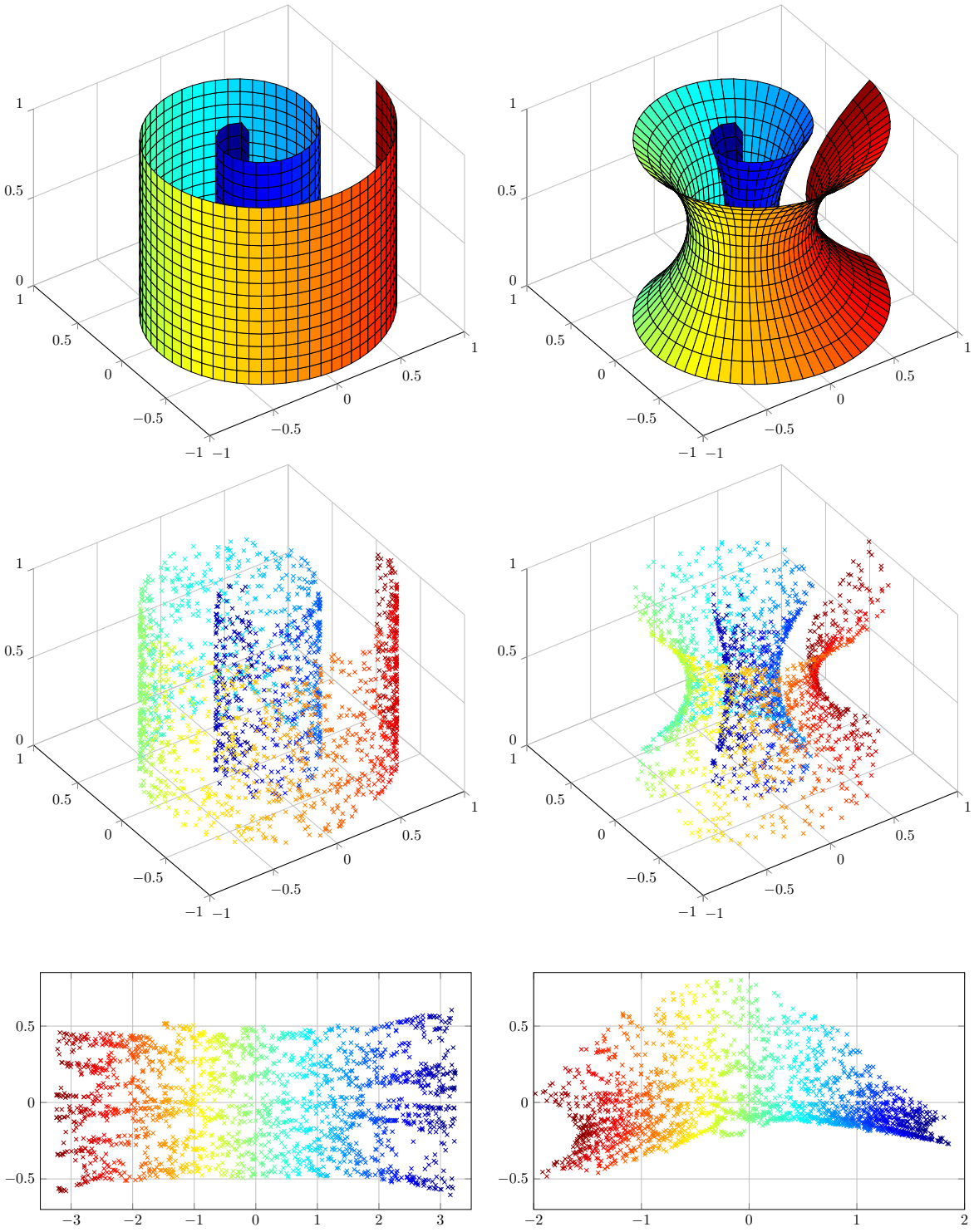

Figure 2.3: *Top:* Surface plot of the "swiss role" *(left)* and "heated swiss role" *(right)*. *Middle:* Both manifolds sampled with 2000 random points each. *Bottom:* Two-dimensional embeddings computed by ISOMAP ($k = 12$).

According to [LV07], the ISOMAP approach is able to recover the underlying structure of developable manifolds, while it fails for non-developable ones. Figure 2.3 now shows surface plots illustrating the two manifolds, where the generating variable y_1 is encoded by the coloring; in the center, both manifolds have been sampled with 2000 random points each; the lower plot finally represents the respective two-dimensional embeddings computed by the ISOMAP algorithm with parameter $k = 12$ for the k-NN graph.

The two-dimensional embedding of the standard swiss role shows that the structure has been unrolled in a satisfying manner. In this case, the graph constructed by ISOMAP is able to perfectly capture the shape of the underlying manifold due to a sufficiently high number of (noiseless) sampling points. It is also clear that a linear method is not able to reproduce this result because of the swiss role's highly nonlinear structure.

The ISOMAP embedding of the heated version does not look completely wrong, however it is also not entirely satisfactory. On the positive side, the data points have been unfolded with respect to the generating variable y_1 encoded by the coloring. On the negative side, the points appear to be clinched with respect to the y_2-variable for y_1 close to 0 (dark red color) and close to 1 (dark blue color). In fact, this characteristic of ISOMAP remains the same for different NN values k. Nevertheless, it should be noted that even the relatively simple example of the heated swiss role poses problems to many other relevant dimension reduction approaches, and only a minority perform well after some parameter tuning, compare [LV07].

Paraboloid

Next, we consider variations of a two-dimensional paraboloid embedded in $\mathbb{R}^3$. The generating function is given by $f_{\text{pa}}^{(\alpha)} : \mathbb{R}^2 \to \mathbb{R}^3$,

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \mapsto \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ (y_1 - 0.5)^2 + (\alpha \cdot (y_2 - 0.5))^2 \end{bmatrix}, \tag{2.26}$$

where $[y_1, y_2]^T$ is uniformly distributed in $[0, 1]^2$. Here, the constant α controls the asymmetry of the paraboloid.

We study the three cases $\alpha = 1, 2, 4$ to show that subtle differences of the underlying manifold can lead to disparate outcomes produced by dimensionality reduction methods. For this purpose, we sample each paraboloid with 2000 points and apply both MDS and ISOMAP (with NN parameter $k = 12$) to reduce the dimension from three to two. The corresponding plots are presented in figure 2.4. Note that the coloring has been chosen to match the y_2 variable of the original dataset in each case.

Let us first have a look at the two-dimensional point sets produced by multidimensional scaling (MDS). Even though the paraboloid is a nonlinear manifold, MDS yields a satisfying result for $\alpha = 1$. This is of course due to the fact that the bending of the chosen segment of the paraboloid is rather low here. For $\alpha = 2$, the manifold is not

unrolled (or flattened) in a desirable way; points close to the "bottom" of the original paraboloid are overlapping. Finally, for $\alpha = 4$, the embedding produced by MDS is not satisfactory at all since it nearly resembles a projection onto the plane spanned by the x_1- and x_3-axes.

The ISOMAP embedding for $\alpha = 1$ is very similar to the one produced by MDS; the manifold is flattened as expected. For $\alpha = 2$, ISOMAP is still able to unroll the paraboloid without heavy distortions. However, the same does not apply to the case of $\alpha = 4$. On the positive side, nearby points in the embedding are also nearby in the original data; in other words, no severe overlapping has been produced by ISOMAP. On the negative side, however, the global structure of the paraboloid has been extremely distorted. Further experiments show that this problem persists also when varying the ISOMAP parameter k. Moreover, for even higher values of α, the corresponding two-dimensional embeddings collapse further and resemble more and more the picture of three straight lines meeting at the origin $(0, 0)$.

The two examples discussed here give some insight into the strong limitations of dimensionality reduction methods. In fact, both toy examples fulfill many favorable conditions that are rarely given in practice: the underlying manifold is perfectly smooth, the sampling is uniform, the data points are noisefree, and the reduction is only required to eliminate a single component. Nevertheless, the outcomes are not fully satisfactory in each considered scenario.

A last remark should be made with respect to the "quality" of the computed embedding. As mentioned in subsection 2.2.1 already, some DR techniques (e.g. as PCA and MDS) feature a precise optimization criterion, i.e., the respective outcome minimizes a particular error measure. Yet, there exists no reasonable approach to define a universally valid error measure for the task of dimensionality reduction. This fact can easily be verified by considering (a random sampling of) the surface of the three-dimensional unit ball, i.e., the two-dimensional unit sphere in $\mathbb{R}^3$. Despite being a smooth 2D-manifold, there is no ideal way of a corresponding two-dimensional embedding. Any such embedding can not reveal the true structure of the sphere, if there is no knowledge of the latent variables and the underlying mapping function.

The low ambient dimension of the above-mentioned examples has been chosen on purpose in order to illustrate certain issues of dimensionality reduction methods visually. However, in most practical scenarios, the ambient dimension is much higher. Indeed, a larger gap between the ambient dimension D and the intrinsic dimension m makes a reliable reduction procedure more beneficial and desirable. In many cases, the intrinsic dimension of a high-dimensional dataset is completely unknown. Recall also that the majority of DR methods rely on an input parameter for the intrinsic dimension. Hence, the first step towards a viable data processing approach requires the determination or, more precisely, the estimation of the intrinsic dimension.

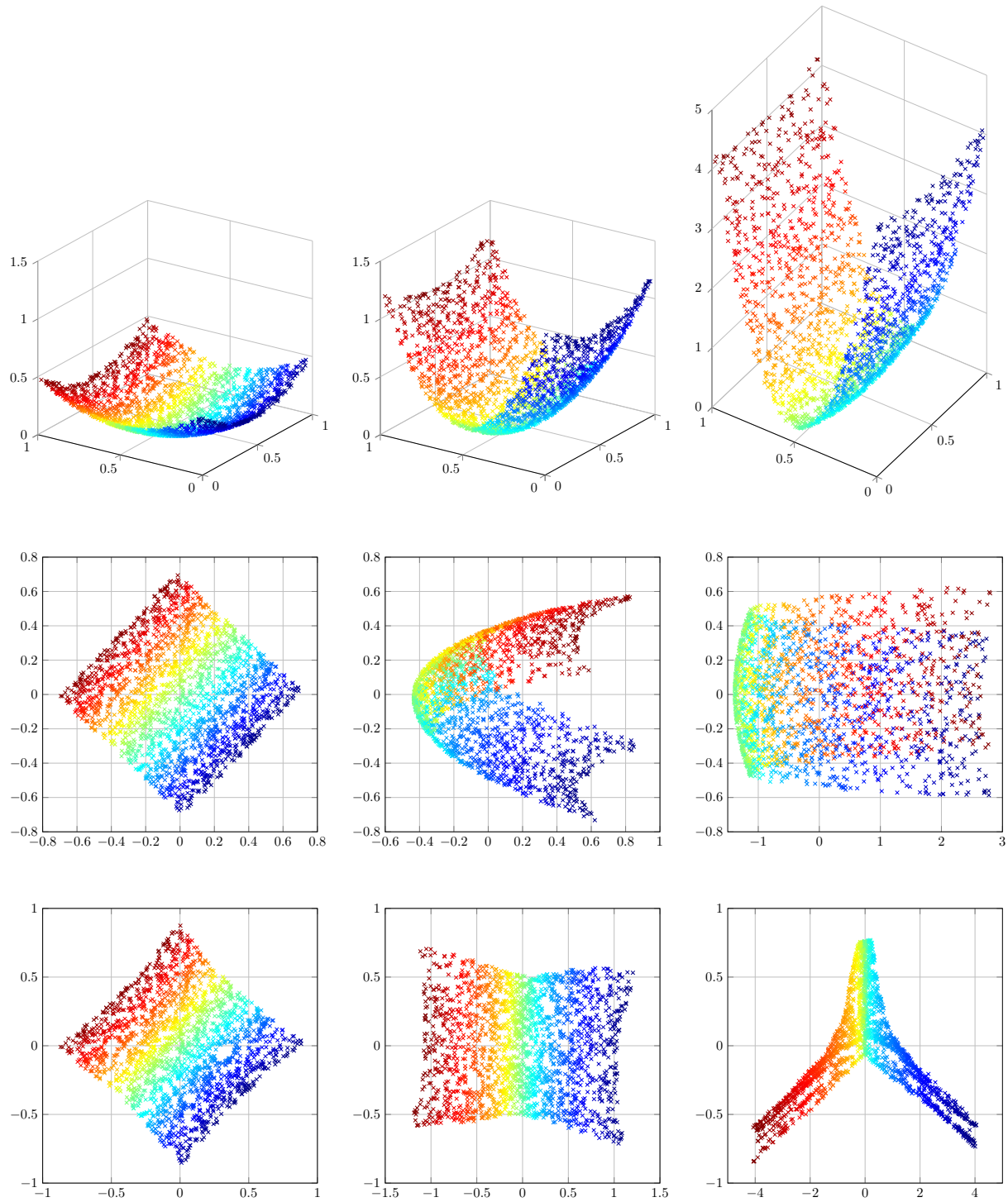

Figure 2.4: *Top (from left to right):* Paraboloid with parameter $\alpha = 1, 2, 4$, sampled with 2000 points. *Middle:* Two-dimensional embeddings computed by MDS. *Bottom:* Two-dimensional embeddings computed by ISOMAP ($k = 12$).

Chapter 3

Intrinsic Dimension Estimation

In this chapter, we discuss the important notion of the *intrinsic dimension* of a set of points. Under the assumption that the points have been generated according to some model, the intrinsic dimension basically coincides with the number of latent variables of the appropriate model. Without an underlying model, the intrinsic dimension can be characterized as the number of variables of a new, lower-dimensional representation of the original data, that is considered sufficient to describe the data according to certain external requirements, e.g. some error measure. As this description already suggests, the intrinsic dimension of a dataset might not be unambiguous, but might rather depend on an error norm or on the purpose of the data processing method. Some concepts of dimension are motivated by fractal geometry, thus, the range of corresponding estimators of the intrinsic dimension is the real interval $[0, D]$, where D is the dimension of the given data. Most dimension reduction methods require a value of the intrinsic dimension as an input parameter, while nevertheless, it could be advantageous in certain cases to combine dimension estimation and reduction in a single approach. In order to get a better understanding of the foundations of different estimation methods, we first consider the most common concepts of dimension.

Before we continue, let us present some notational conventions that we make use of throughout this chapter. All descriptions in table 3.1 are effective unless noted otherwise.

3.1 Concepts of Dimension

In mathematics, the term "dimension" is employed in different areas with different meanings. The most well-known and elementary definition is the dimension of a vector space determined by the cardinality of its basis. This includes the Euclidean space of dimension n, denoted by $\mathbb{R}^n$. Since a manifold is defined as a topological space that is locally homeomorphic to the Euclidean space of some fixed dimension n, the corresponding dimension of the manifold is also given by n in a straightforward way.

In our context, problems arise when we consider a countable (or usually finite) set

symbol	explanation
$\mathbb{R}^n$	Euclidean space of dimension n
$\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}$	bold symbols denote vectors, typically elements of $\mathbb{R}^n$ for some given n
$\|\boldsymbol{x}\|$	Euclidean norm: $\|\boldsymbol{x}\| = \|\boldsymbol{x}\|_2 = \sqrt{x_1^2 + \ldots + x_n^2}$
$B_r(\boldsymbol{x})$	ball with radius r centered at $\boldsymbol{x}$
D	dimension of ambient space (i.e., original data space)
m	intrinsic dimension of data (or number of latent variables)
N	number of observed data points
$\mathcal{X}$	set of observed data points: $\mathcal{X} = \{\boldsymbol{x}_{i=1,\ldots,N}\} \subset \mathbb{R}^D$
$\boldsymbol{x}_i^{(j)}$	jth nearest neighbor of point $\boldsymbol{x}_i$
$T_j(\boldsymbol{x}_i)$	Euclidean distance of $\boldsymbol{x}_i$ to its jth nearest neighbor: $T_j(\boldsymbol{x}_i) = \left\|\boldsymbol{x}_i - \boldsymbol{x}_i^{(j)}\right\|$

Table 3.1: Notational conventions for Chapter 3. The last two expressions are used if and only if we are referring to a fixed and finite set of points. The term $\boldsymbol{x}_i^{(j)}$ might not be well-defined if there exist two points at the same distance from $\boldsymbol{x}_i$. In this case, we implicitly assume some fixed ordering of the nearest neighbors. The exact kind of this ordering is of no relevance.

of points sampled from some unknown manifold. The topological dimension, which will be introduced below, of any finite set equals zero, hence we need to study alternative concepts which brings us to fractal dimensions. For sufficiently well-behaved sets, these fractal dimensions coincide with the standard topological dimension. However for the so-called fractals, which are basically characterized by self-similarity and a recursive construction process, the fractal dimensions usually evaluate as non-integer real numbers larger than the topological dimension. Furthermore, the nature of their definition often leads to a practical approach to assign a meaningful value (of dimension) to a finite point set.

In the following, we first present the topological dimension, also known as Lebesgue covering dimension, then the Hausdorff dimension being the most prominent example of a fractal dimension, before we consider the box-counting and the correlation dimension, which can both be subsumed in the more general concept of the q-dimension. Parts of this section are based on [LV07] and [Fal03], a nice introduction to the topic can also be found in [Cut93].

3.1.1 Lebesgue covering dimension

One of the most fundamental and well-understood concepts is the Lebesgue covering dimension, also called *topological dimension* or just *covering dimension*. It is defined

with respect to a given topological space $(\mathcal{Y}, \tau)$, that is a set of points $\mathcal{Y}$ together with a collection τ of subsets of $\mathcal{Y}$ called open sets. For any subset $S \subset \mathcal{Y}$, a *covering* of S is defined as a family $\mathcal{C}$ of open sets whose union contains S. For any covering $\mathcal{C}$, a *refinement* $\mathcal{C}'$ of $\mathcal{C}$ is another covering such that each set of $\mathcal{C}'$ is contained in some set of $\mathcal{C}$.

Now the *Lebesgue covering dimension of* $S \subset \mathcal{Y}$ is defined as the smallest integer D_L, such that every covering $\mathcal{C}$ of S has a refinement $\mathcal{C}'$, for which each point of S is contained in at most $D_L + 1$ sets of $\mathcal{C}'$. If such an integer does not exist, the covering dimension is infinite.

Provided that $\mathcal{Y}$ is a separated space (also known as *Hausdorff space*), it is easy to see that any finite set has a topological dimension of zero, since for each point, an open neighborhood containing no other points can be found. On the other hand, there are also unions of uncountable closed sets with dimension zero, such as the famous Cantor set.

The geometric idea behind this notion of dimension can be illustrated in the Euclidean space by considering simple one- and two-dimensional examples like a line segment, a circle, or a disk. For example, any covering of a circle can be refined such that each set of the refinement contains only an open arc of the circle. Thus, each point of the circle is contained in no more than two sets of the refinement and consequently, the dimension of the circle is one.

As mentioned above, the Lebesgue covering dimension, being a very general topological concept, is of limited use when it comes to analyzing certain intricate point sets. The following fractal dimensions provide the necessary flexibility.

3.1.2 Hausdorff dimension

The Hausdorff dimension is defined with respect to a metric space $(\mathcal{Y}, d_{\mathcal{Y}})$. For a set $S \subset \mathcal{Y}$, a countable family $\mathcal{C} = \{T_i\}$ of open sets is called an ϵ-covering of S, if the union of all T_i contains S, and the diameter of all T_i is bounded by ϵ. Here, the diameter of T_i is defined as $\operatorname{diam}(T_i) := \sup\{d_{\mathcal{Y}}(u, v) \mid u, v \in T_i\}$. Now, given some real number $\alpha \geq 0$, the *α-Hausdorff measure of* S is defined as

$$H^{\alpha}(S) = \lim_{\epsilon \to 0} \inf_{\mathcal{C}} \left\{ \sum_{T_i \in \mathcal{C}} \operatorname{diam}(T_i)^{\alpha} : \mathcal{C} \text{ is an } \epsilon\text{-covering of } S \right\}. \tag{3.1}$$

It should be recalled here that Hausdorff measures generalize the concept of n-dimensional volume, i.e. the n-dimensional Lebesgue measure. In particular, for any Borel subset $S \subset \mathbb{R}^n$, we have (see [Fal03])

$$H^n(S) = c_n \cdot \operatorname{vol}_n(S), \tag{3.2}$$

where the constant c_n is the volume of the n-ball of diameter 1.

Since it can be shown that the following infimum and supremum exist and coincide, the *Hausdorff dimension of* S is now defined as

$$D_H(S) = \inf\{\alpha : \ H^\alpha(S) = 0\} = \sup\{\alpha : \ H^\alpha(S) = \infty\}. \tag{3.3}$$

Thus, the Hausdorff dimension is given by the precise value of α, where the Hausdorff measure H^α "jumps" from infinity to zero. It can be shown that this jump occurs at the same place when the measure is slightly modified. Instead of all ϵ-coverings of S, it suffices to consider only all coverings with balls of radius less than or equal to ϵ. This leads to the following intuitional view: consider a sufficiently well-behaved set S and each covering of S with such balls. Let N_ϵ be the smallest number of balls of this covering. Then, as ϵ approaches zero, the number N_ϵ grows as ϵ^{-d}, where d is the Hausdorff dimension.

Since it will be sufficient for our further considerations, from now on, we restrict ourselves to the case $\mathcal{Y} = \mathbb{R}^n$. As noted in [Fal03], the Hausdorff dimension features the following important properties:

- *Monotonicity:* For each $S \subseteq T \subset \mathbb{R}^n$: $D_H(S) \leq D_H(T)$.
- *Countable stability:* $D_H\left(\bigcup_{i=1}^{\infty} S_i\right) = \sup_{1 \leq i < \infty} D_H(S_i)$.
- *Countable sets:* For each countable set $S \subset \mathbb{R}^n$: $D_H(S) = 0$.
- *Geometric invariance:* $D_H(f(S)) = D_H(S)$, if f is an isometry or an affine transformation of $\mathbb{R}^n$.
- *Open sets:* For each open subset $S \subset \mathbb{R}^n$: $D_H(S) = n$.
- *Smooth manifolds:* For each continuously differentiable m-dimensional submanifold S of $\mathbb{R}^n$: $D_H(S) = m$.
- *Bi-Lipschitz invariance:* For $S \subset \mathbb{R}^n$ let $f : S \to \mathbb{R}^m$ be a bi-Lipschitz transformation, i.e., there exist two positive constants $c_1 \leq c_2$ with $c_1\|\boldsymbol{x} - \boldsymbol{y}\| \leq \|f(\boldsymbol{x}) - f(\boldsymbol{y})\| \leq c_2\|\boldsymbol{x} - \boldsymbol{y}\|$ for all $\boldsymbol{x}, \boldsymbol{y} \in S$. Then $D_H(f(S)) = D_H(S)$.

Intuitively, all those properties might seem natural and desirable for any notion of dimension. Nevertheless, we will see that the *box-counting dimension* introduced below, which is closely related to the Hausdorff dimension, violates even basic concepts, e.g. the *countable stability* and *countable sets* properties.

In the following, we present two fractal dimension definitions which are the foundation for many important methods in practical dimension computation and estimation.

3.1.3 The box-counting dimension

The intuition behind the box-counting dimension, the correlation dimension, and the Hausdorff dimension is in fact quite similar. Let $N_\epsilon(S)$ denote the number of "volume-like" elements scaling with parameter ϵ, where each element is associated with some point of the d-dimensional set S. Then, for small ϵ, we expect the measurement to obey an exponential dependence on d:

$$N_\epsilon(S) \sim \epsilon^{-d}. \tag{3.4}$$

After applying the logarithm and letting $\epsilon \to 0$, we yield

$$d \sim \lim_{\epsilon \to 0} \frac{\log N_\epsilon(S)}{-\log \epsilon}. \tag{3.5}$$

In fact, this relationship is already very close to the precise definition of the box-counting dimension, also known as *Minkowski-Bouligand dimension* or *capacity dimension*. Given some non-empty bounded subset $S \subset \mathbb{R}^n$, define $N_\epsilon(S)$ as the smallest number of sets of diameter at most ϵ covering S. Then, the *lower* and *upper box-counting dimensions of* S are defined as

$$D^-_{\text{box}}(S) = \liminf_{\epsilon \to 0} \frac{\log N_\epsilon(S)}{-\log \epsilon}, \qquad D^+_{\text{box}}(S) = \limsup_{\epsilon \to 0} \frac{\log N_\epsilon(S)}{-\log \epsilon}. \tag{3.6}$$

Finally, if the two limit values coincide, their common value

$$D_{\text{box}}(S) = \lim_{\epsilon \to 0} \frac{\log N_\epsilon(S)}{-\log \epsilon} \tag{3.7}$$

is called the *box-counting dimension of* S.

So far, the naming might not seem appropriate, since the definition does not involve particular objects such as boxes. For this reason, it must be noted that the box-counting dimension can be defined equivalently using any of the following quantities for $N_\epsilon(S)$:

(i) the smallest number of closed ϵ-balls covering S,

(ii) the smallest number of ϵ-cubes covering S,

(iii) the number of ϵ-mesh cubes intersecting S,

(iv) the largest number of disjoint ϵ-balls with centers in S.

The quantities (ii) and (iii) are frequently used in practical dimension estimation. For example, when using (iii), the set S is covered with ϵ-meshes for decreasing values of ϵ and the number of boxes overlapping S are counted for each ϵ. The dimension can then be estimated as the logarithmic rate at which $N_\epsilon(S)$ increases for $\epsilon \to 0$.

Furthermore, when it comes to estimating the Hausdorff dimension of a set, for which only a finite number of sample points is available, the box-counting dimension is often the method of choice due to its simplicity. In fact, the Hausdorff and the box-counting dimension are equal for many sufficiently regular subsets of $\mathbb{R}^n$; however, one can show

that

$$D_H(S) \leq D_{\text{box}}(S) \qquad \text{for each } S \subset \mathbb{R}^n, \tag{3.8}$$

and numerous examples are known where the two dimensions actually differ. Consider e.g. the countable set S_1 of all rational numbers in the unit interval $[0, 1]$. It can easily be derived that $D_{\text{box}}(S_1) = 1$, while $D_H(S_1) = 0$, since S_1 is countable. Another rather counterintuitive case is represented by the set $S_2 = \left\{0, 1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \dots\right\}$, which has a single limit point; nevertheless, its box-counting dimension is $D_{\text{box}}(S_2) = \frac{1}{2}$.

These results show that the box-counting dimension violates both the *countable stability* and the *countable sets* properties of the Hausdorff dimension, which is however not a problem in common practical applications. To the contrary, one often seeks to analyze the dimension of a set or manifold which is represented by only finitely many points, where the Hausdorff dimension itself is not helpful at all.

3.1.4 The correlation dimension

The correlation dimension has been introduced by Grassberger and Procaccia in [GP83] in the context of time series and attractors. In a nutshell, an attractor is a set of numerical values, often with either some manifold-like or fractal-like structure, acting like a local or global equilibrium for an underlying system, meaning that the underlying system tends to evolve towards the attractor. Even though the correlation dimension can be defined in a more general way, we prefer the following quite intuitive approach.

Consider a countable set $S = \{\boldsymbol{s}_{i=1,\dots,\infty}\} \subset \mathbb{R}^n$ and define the so-called *correlation sum of* S,

$$C_\epsilon(S) = \lim_{K\to\infty} \frac{2}{K(K-1)} \sum_{1\leq i<j\leq K} H\left(\epsilon - \|\boldsymbol{s}_i - \boldsymbol{s}_j\|\right), \tag{3.9}$$

where $H : \mathbb{R} \to \mathbb{R}$ is the Heaviside step function defined by

$$H(x) = \begin{cases} 0 & \text{if } x < 0, \\ 1 & \text{if } x \geq 0. \end{cases} \tag{3.10}$$

The correlation sum describes the probability that some arbitrary pair of points $(\boldsymbol{s}_i, \boldsymbol{s}_j)$ features a distance of no more than ϵ. Now, analogously as for the box-counting dimension, if the corresponding limit inferior and limit superior coincide, the *correlation dimension of* S is defined as

$$D_{\text{cor}}(S) = \lim_{\epsilon\to 0} \frac{\log C_\epsilon(S)}{\log \epsilon}. \tag{3.11}$$

For the intuition behind the correlation dimension, consider a fixed point $\boldsymbol{s}_i$ and its ϵ-neighborhood $B_\epsilon(\boldsymbol{s}_i)$. Then, the growth rate of the number of points $\boldsymbol{s}_j$ within $B_\epsilon(\boldsymbol{s}_i)$ for increasing ϵ should be exponential with respect to the underlying dimension. Thus,

while the box-counting dimension is related with a global covering of the complete point set, the correlation dimension relies on a local analysis of isolated points.

To estimate the correlation dimension in practice, it suffices to compute all (or a large number of) pairwise inter-point distances and subsequently approximate the correlation sum $C_\epsilon(S)$ for decreasing values of ϵ. The challenging part here — as with the box-counting dimension — is to determine an appropriate range of values for ϵ and to yield a proper logarithmic rate for $\epsilon \to 0$. Details for this issue will be discussed in the upcoming section.

Finally, both notions of dimension can be united in a more general concept that we now present in aggregate form.

3.1.5 The q-dimension

The q-dimension is usually attributed to Rényi (see [Rén59]) and hence it is sometimes also referred to as *Rényi dimension*. An equivalent definition has also been provided in [HP83] by Hentschel and Procaccia, again in the context of chaotic dynamical systems and their attractors. A review of different definitions can be found in [Pes93].

Instead of for some subset of $\mathbb{R}^n$, the q-dimension is defined more generally for a given Borel probability measure μ on a metric space $\mathcal{Y}$. For each $q \geq 0$ and $\epsilon > 0$, first define the correlation integral

$$C_{q,\epsilon}(\mu) = \int_{\mathcal{Y}} \left(\mu(\bar{B}_\epsilon(y))\right)^{q-1} d\mu(y), \tag{3.12}$$

where $\bar{B}_\epsilon(y)$ denotes the closed ball of radius ϵ with center y. Next, for $q \neq 1$, the *lower* and *upper q-dimensions of μ* are given by

$$D_q^-(\mu) = \liminf_{\epsilon \to 0} \frac{\log C_{q,\epsilon}(\mu)}{(q-1)\log \epsilon}, \qquad D_q^+(\mu) = \limsup_{\epsilon \to 0} \frac{\log C_{q,\epsilon}(\mu)}{(q-1)\log \epsilon}. \tag{3.13}$$

In case the two limit values coincide, their common value

$$D_q(\mu) = \lim_{\epsilon \to 0} \frac{\log C_{q,\epsilon}(\mu)}{(q-1)\log \epsilon} \tag{3.14}$$

is called the *q-dimension of μ*.

In our context, the most important properties of the q-dimension are the following. First, for a given metric space $\mathcal{Y}$ and a probability measure μ on $\mathcal{Y}$, consider a bounded and μ-measurable set $S \subset \mathcal{Y}$. Then, using the corresponding correlation integral

$$C_{q,\epsilon}(\mu, S) = \int_S \left(\mu\left(\bar{B}_\epsilon(y)\right)\right)^{q-1} d\mu(y), \tag{3.15}$$

we yield the q-dimension of S, $D_q(\mu, S)$. Now, one can easily derive that $D_0(\mu, S)$

corresponds to the box-counting dimension and $D_2(\mu, S)$ corresponds to the correlation dimension.

It is also noteworthy that after taking the limit $q \to 1$ and applying de l'Hospital's rule, the result $D_1(\mu, S)$ is referred to as *information dimension*. In practice however, the information dimension does not play a vital role since it is generally complicated to estimate (see [LV07]).

Finally, the following inequality has already been shown in [HP83]:

$$\text{For all } 0 \leq q_1 < q_2 < \infty : \quad D_{q_2}(\mu) \leq D_{q_1}(\mu), \tag{3.16}$$

which naturally implies $D_2(\mu) \leq D_1(\mu) \leq D_0(\mu)$.

3.1.6 Further notions of dimension

Besides the widespread concepts of dimension discussed above, there exist many more, most of which come with subtle differences. Before moving on, we would like to name just a few rather well-known variants without going into further details.

Approaches based on the open covering of the given point set include the *Assouad dimension* and the *Aikawa dimension*. The first has been introduced by Assouad in [Ass79] and is based on a covering of the set with open balls. An in-depth review including alternative definitions can be found in [Luu98]. The second is due to Aikawa (see [Aik91]) and based on integrals of the distance function. In [LT13a] however, it has been shown that for all subsets of $\mathbb{R}^n$, the Assouad and the Aikawa dimension coincide.

In the context of the theory of attractors, another commonly used concept is the *Kaplan-Yorke dimension*, also known as *Lyapunov dimension*. Since this topic is not the focus of our work, we refer the reader interested in deeper insights to the original publication [KY79] as well as the nice surveys in [FOY83] and [Ott02].

3.2 Estimation of the Intrinsic Dimension

Just as the concept of dimension, the notion of the intrinsic dimension of a point set has been established in various research fields with slightly different meanings. The term "intrinsic dimension" is sometimes imputed to Bennett (cf. [Ben65]), while the basic idea has certainly been around much earlier, e.g. when Peano introduced a continuous mapping from the unit interval onto the unit square, a so-called space-filling curve, in [Pea90].

Due to the multitude of characterizations and approaches, we do not aim for a comprehensive summary of all existing concepts of intrinsic dimension. Reviews of estimation methods can be found in [Cam03] or [MM10] amongst others. Furthermore, [CCCR15] provides a very recent overview of successful methods, while a more theoretical approach to the topic is presented in [Pes08].

In the following subsection, we propose several ways to categorize estimation methods, we highlight the most relevant examples with their associated modifications, and we discuss the corresponding advantages and drawbacks. Subsequently, we analyze certain carefully selected approaches for dimension estimation in more detail. From now on, when appropriate, we use the acronym IDE for *intrinsic dimension estimation*.

3.2.1 Classification and characteristics of IDE methods

The underlying model

What is the intrinsic dimension of a given dataset $\mathcal{X} \subset \mathbb{R}^D$? There are at least two different answers to this question, depending on the underlying assumptions. The first model is based on some generating function $f : \mathbb{R}^m \to \mathbb{R}^D$ and assumes that the observed data $\mathcal{X} = \{\boldsymbol{x}_i\}$ has been generated from the set of latent variables $\{\boldsymbol{y}_i\} \subset \mathbb{R}^m$ via $\boldsymbol{x}_i = f(\boldsymbol{y}_i) + \boldsymbol{\epsilon}_i$, where $\boldsymbol{\epsilon}_i$ denotes the noise term. In this very general setting, the intrinsic dimension is the dimensionality of the latent space, m. The second model is a geometric approach and just assumes that the data $\mathcal{X} \subset \mathbb{R}^D$ is situated on or, in the noisy case, close to some unknown manifold. Here, the intrinsic dimension clearly is the corresponding manifold dimension.

The latent variable model can usually not be applied without any further assumptions on the generating function f. To see this, just let f be a space-filling curve, e.g. a generalized Hilbert curve, mapping the unit interval $[0, 1]$ onto the unit cube $[0, 1]^n$. There is no intuitive reasoning according to which the observed variables $\{\boldsymbol{x}_i\}$ of this model should have an intrinsic dimension of $m = 1$.

Certainly, any knowledge about the generating function f can potentially be used both to give a more precise definition of the intrinsic dimension and to choose a specific estimation method. However, in many cases, not only f is completely unknown or highly nonlinear, but also the measurement errors have an enormous impact on the observed data. Hence, numerous approaches ([CH04b], [HA05], [BL05], [RL06], [MM10]) prefer the manifold model as definition of the intrinsic dimension.

Next, we discuss some specific characteristics that allow to classify estimation methods more precisely. It must be noted though that several approaches can not be categorized into the following scheme in a completely unambiguous way. Nevertheless, it is still useful to keep in mind the following distinctive features. Our considerations are partly based on [LV07] and [CCCR15].

Global and local methods

Methods that compute a single estimate using the entire dataset at once are called *global* methods. Popular examples are Principal Component Analysis (PCA) and Multidimensional Scaling (MDS). In contrast, *local* methods assign a different estimate to local patches of the data or even to each single point. If required, local results are then combined into a global estimate, using averaging or voting or some more sophisticated

technique. A global method can be made local by providing a suitable way to extract local subsets of the data and then applying the original method to each subset separately.

Global methods can usually be successfully applied for datasets with a simple underlying structure, such as an affine subspace or some smooth manifold with a constant, a priori given curvature. However, for more complex manifolds with non-constant curvature or highly non-linear features, or also for a mixture of different manifolds, local methods are in general much better suited. This seems natural regarding the characterization of a manifold as a locally smooth and low-dimensional object.

While the discrimination between global and local methods only refers to the associated output mode, it might also be beneficial to consider the precise technique, which can be either global, local, or multilevel. PCA and MDS are also global in this regard, since they work with all data points at once. In contrast, the estimator by Grassberger and Procaccia (see subsection 3.2.2) yields a global estimate via a multilevel approach, considering the dataset at different scales. On the other hand, the multiscale SVD (see below) method uses also different, relatively small scales to get multiple local estimates. Eventually, most nearest neighbor methods rely just on a single, local scale for their local estimation results.

Multilevel techniques clearly provide the highest flexibility, since they allow for the examination of global as well as different local structures. On the other hand, the flexibility of local and especially multilevel methods usually comes at the expense of a higher complexity, e.g. regarding the algorithmic implementation, the tuning of multiple parameters, or cost complexity. Nevertheless, nearly all recent approaches rely on either local or multilevel schemes and frequently deliver more precise estimation results than classical global variants.

Projective methods

Projective approaches explicitly construct a mapping that projects the given data into an appropriate space. The intrinsic dimension is either the dimension of the image space or it is extracted as a property of the projection itself. Many of these particular methods have been primarily designed as dimension reduction procedures in the context of data exploration or visualization and require the intrinsic (target) dimension as input parameter. Nevertheless, modifications and extensions have been presented that allow for an estimation of the ID.

The majority of all projective methods are based on one of two well-established techniques with a long history going back to the beginning of the 20th century: Principal Component Analysis (PCA, see [Jol02]) and Multidimensional Scaling (MDS, see [BG05]). While both variants do not come with a trivial standard approach to compute the intrinsic dimension, still many estimation methods, ranging from simple to sophisticated, rely on the one or the other, which is why we present the two concepts in a more detailed manner.

PCA can be interpreted — from a statistical viewpoint — as maximization of the

preserved variance in the data or — from an approximation viewpoint — as minimization of the reconstruction error. In a nutshell, PCA can be performed via the following steps. First, center all data points $\boldsymbol{x}_{i=1,\ldots,N} \in \mathbb{R}^D$ via subtracting their mean value to yield the shifted points

$$\bar{\boldsymbol{x}}_i = \boldsymbol{x}_i - \left(\frac{1}{N} \sum_{j=1}^{N} \boldsymbol{x}_j \right). \tag{3.17}$$

Assume now that the shifted data $\bar{\boldsymbol{x}}_{i=1,\ldots,N}$ are arranged in an $(N \times D)$-matrix X. Next, consider the covariance matrix defined by

$$W = X^T \cdot X, \tag{3.18}$$

which is positive-semidefinite and symmetric. Consequently, the eigenvalue decomposition of W can be written as

$$W = S \Lambda S^T, \tag{3.19}$$

where the diagonal matrix $\Lambda = \mathrm{diag}\,(\lambda_1, \ldots, \lambda_D)$ contains the eigenvalues of W in descending order $\lambda_1 \geq \ldots \geq \lambda_D \geq 0$ and the columns of S are the corresponding orthonormal eigenvectors. The core part of the principal component analysis is now the transformation matrix S. The columns of S are called "components", and the projection on the first few components (associated with the largest eigenvalues) preserves the maximum of variance in the data. Now, the proper selection of the number m of relevant principal components is equivalent to the estimation of the intrinsic dimension. One possible approach is to fix some small $\epsilon > 0$ and determine the smallest $m \leq D$ such that

$$\frac{\sum_{k=1}^{m} \lambda_k}{\sum_{k=1}^{D} \lambda_k} \geq 1 - \epsilon. \tag{3.20}$$

By keeping only the first m components, the fraction $(1 - \epsilon)$ of the global variance is preserved in the reduced data.

Principal Component Analysis is a purely linear transformation. Therefore, it is generally not able to discover any nonlinear structure and its usefulness is rather limited, unless it is modified or combined with other techniques.

In practice, for very high-dimensional data residing in $\mathbb{R}^D$ with low intrinsic dimension $m \ll D$, PCA is occasionally applied as a first step to project the data into the best-approximating d-dimensional space with $m < d \ll D$, where subsequently, a more advanced scheme is utilized to recover the true ID value m from the data embedded in the much smaller ambient space.

Many different modifications of the PCA method exist. A local variant has first been presented in [FO71], where PCA is applied in local hyperspheres of varying sizes. An extension of this variant using optimally topology preserving maps for a superior handling of noisy data is introduced in [BS98]. In [SSM98], the so-called *kernel PCA* was coined as a non-linear generalization of PCA, which now takes place in an arbitrarily high-dimensional feature space determined by the chosen kernel function. The authors claim

that fewer nonlinear principal components are required to get the same performance in classification tasks as with standard PCA. However, on the downside, the reconstruction of the original data from the components is highly non-trivial.

Probabilistic variants have been proposed in [TB99] and [Bis99], where the PCA is reformulated as a maximum likelihood problem. This allows for a combination of multiple PCA models into a probabilistic mixture model. Further, the *Bayesian PCA* in [Bis99] introduces a way to automatically derive the number of latent variables (i.e., the intrinsic dimension) under certain model assumptions. An interesting generalization has been established in [CDS02], where the Gaussian distribution in the underlying PCA model is replaced by the exponential family, allowing for better suited models in the case of integer or binary data. A recent refinement of this approach can also be found in [LT13b].

An interesting extension, the so-called *sparse PCA*, is presented in [ZHT06], the corresponding probabilistic variant in [GD09]. While each standard principal component usually depends on all given input variables, the sparse PCA imposes the constraint that the sparse PCs only depend on a few input variables, thus allowing for a more straightforward interpretation of the results in certain scenarios.

Finally, a promising approach named *multiscale SVD* has been published in [LMR12, LMR11]. Based on tools from multiscale geometric measure theory and harmonic analysis, the authors implement a carefully constructed model with particular assumptions on the manifold's curvature, the sampling, and the imposed noise level. They show that under these conditions, for an m-dimensional manifold and for an appropriate local scale $r > 0$, the corresponding singular values of the local SVDs can be separated into m values growing like r, while the remaining ones grow like r^2. This gap in the singular values is finally utilized in a multiscale approach to determine the intrinsic dimension m. Their algorithm yields very good results for artificial datasets, in particular for high-dimensional unit cubes and spheres tainted with relatively high levels of Gaussian noise. However, in experiments for real world datasets presented in [LMR12, LMR11], the multiscale SVD method frequently produces estimates much lower than all competing approaches, and no attempt of an explanation for this behavior is provided.

The second category consists of methods based on multidimensional scaling (MDS). This term describes a concept rather than a specific procedure. The key idea of MDS is to preserve most of the pairwise similarities (or dissimilarities) between data points while embedding them into a lower-dimensional space. To achieve this, an associated loss function (called *stress* or *strain*) measuring the dissimilarity between the original and projected similarity matrix, is minimized. Different definitions of loss functions lead to different variants of MDS. Generally, one distinguishes between metric and non-metric MDS. While metric versions seek to preserve the exact values of the similarities as well as possible, non-metric versions are linked to the ranking of the similarities. Further details can be found in [CC00, BG05], a nice historical introduction is provided in [GB14].

The most prominent example of metric MDS is due to Torgerson (see [Tor52]) and it is usually referred to as *classical multidimensional scaling* or sometimes as *principal*

coordinates analysis. This approach can also be thought of as a "transposed analogue" of PCA. Classical MDS does not require the precise coordinates of the input points, but rather relies on the matrix of pairwise squared Euclidean distances between them. As above, let the centered data points $\bar{\boldsymbol{x}}_{i=1,\dots,N}$ be arranged in an $(N \times D)$-matrix X and let further $Q = [q_{ij}]$ with $q_{ij} = \|\bar{\boldsymbol{x}}_i - \bar{\boldsymbol{x}}_j\|^2$ be the squared distance matrix. Now, consider the Gram matrix defined by

$$B = X \cdot X^T, \tag{3.21}$$

which is positive-semidefinite and symmetric. Note that B can be rewritten as

$$B = -\frac{1}{2} JQJ, \tag{3.22}$$

where $J = \boldsymbol{I}_N - \frac{1}{N}\mathbf{1}_N \cdot \mathbf{1}_N^T$ is a centering matrix, $\boldsymbol{I}_N$ the N-dimensional identity matrix, and $\mathbf{1}_N$ the N-dimensional vector containing ones everywhere. Finally, let

$$B = S\Lambda S^T \tag{3.23}$$

be the eigenvalue decomposition with $\Lambda = \operatorname{diag}(\lambda_1, \dots, \lambda_N)$ and the orthonormal matrix S containing the eigenvectors as columns. Here, at most D of the N eigenvalues λ_i can be positive, while the remaining eigenvalues are zero. The final dimension reduction step corresponds to the selection of the number m of components to be kept. The reduced data is then given by

$$\hat{X} = S_m \Lambda_m^{1/2}, \tag{3.24}$$

where the $(m \times m)$-matrix Λ_m contains the largest eigenvalues on its diagonal and the $(N \times m)$-matrix S_m holds the associated eigenvectors.

It is straightforward to prove that both PCA and classical MDS are linear reduction methods and yield equivalent results, see e.g. [LV07].

Classical MDS minimizes the so-called *strain loss function* given by

$$f_{\text{MDS}}^{(1)}(\hat{\boldsymbol{x}}_1, \dots, \hat{\boldsymbol{x}}_N) = \left(\frac{\sum_{i,j} \left(b_{ij} - \langle \hat{\boldsymbol{x}}_i, \hat{\boldsymbol{x}}_j \rangle \right)^2}{\sum_{i,j} b_{ij}^2} \right)^{\frac{1}{2}}, \tag{3.25}$$

where $\hat{\boldsymbol{x}}_i$ are the output data and b_{ij} are the entries of the matrix B.

It must further be noted that classical MDS is also utilized in situations where first the points $\bar{\boldsymbol{x}}_{i=1,\dots,N}$ are unknown and second the given matrix $Q = [q_{ij}]$ corresponds to pairwise dissimilarities, but not Euclidean distances between the data points. This case must be treated in a slightly different way, since some of the eigenvalues $\lambda_1, \dots, \lambda_N$ can now be negative.

As mentioned above, many different variants of MDS exist. In general — except for classical MDS — the associated function to be minimized is called *stress function* due to Kruskal, who proposed the following stress function prototype for the output

configuration $\hat{\boldsymbol{x}}_1, \ldots, \hat{\boldsymbol{x}}_N$ in [Kru64a, Kru64b]:

$$f_{\mathrm{MDS}}^{(2)}\left(\hat{\boldsymbol{x}}_1, \ldots, \hat{\boldsymbol{x}}_N\right) = \left(\frac{\sum_{i<j} \left(d_{ij} - \delta(q_{ij})\right)^2}{\sum_{i<j} d_{ij}^2} \right)^{\frac{1}{2}}. \tag{3.26}$$

Here, $d_{ij} = d(\hat{\boldsymbol{x}}_i, \hat{\boldsymbol{x}}_j)$ represents an arbitrary distance function, e.g. the Minkowski distance of order p, while $\delta(q_{ij})$ denotes some general transformation of the input dissimilarities q_{ij}.

Even in the most basic setting, where $d_{ij} = \|\hat{\boldsymbol{x}}_i - \hat{\boldsymbol{x}}_j\|$ denotes the Euclidean distance and also $\delta(q_{ij}) = q_{ij}$ represent the measured Euclidean distances, the solution of the minimization problem with loss function (3.26) has no analytical solution (like classical MDS) and thus requires for example a steepest descent method. However, as opposed to the classical version, most variants are able to recover underlying nonlinear structures in the data.

First steps using multidimensional scaling to determine the intrinsic dimensionality of signal collections are due to Shepard and Carroll [SC66], Kruskal, Bennett [Ben69], Trunk [Tru68], and others, see [CA74] and the references therein.

Popular and successful methods primarily used in dimension reduction include *Sammon's mapping* [Sam69], *Kohonen's self-organizing maps* [Koh89], and more recent approaches like *curvilinear component analysis* (CCA, [DH97]), *locally linear embedding* (LLE, [RS00, RS03]) and ISOMAP [TdSL00]. Detailed descriptions and evaluations are presented in [LV07]. The most promising among these approaches are probably ISOMAP and LLE.

Simply speaking, ISOMAP combines classical MDS with graph distances, which on their part approximate geodesic distances. This modification results in a nonlinear method. The graph distances are computed as follows. First, a weighted nearest neighbor graph G is constructed for all input points, which is either a k-NN or an ϵ-ball graph. More precisely, given $\boldsymbol{x}_i$, the point $\boldsymbol{x}_j$ is connected to $\boldsymbol{x}_i$ either if it is among the k nearest neighbors of $\boldsymbol{x}_i$, or if their distance fulfills $\|\boldsymbol{x}_i - \boldsymbol{x}_j\| < \epsilon$. In each case, the edge weight equals $\|\boldsymbol{x}_i - \boldsymbol{x}_j\|$. Now, the graph distances are evaluated via constructing the shortest path between two points in G and summing up the corresponding weights. Finally, the classical MDS procedure is applied to the matrix of graph distances.

ISOMAP is a great improvement over classical MDS. While it inherits the advantage of algorithmic and computational simplicity, it is also able to successfully recover certain nonlinear manifolds, for example the famous "swiss role", as shown already in subsection 2.2.2. However, recall that ISOMAP suffers from the following two main issues. The class of recoverable nonlinear manifolds is restricted to the *developable* manifolds. Basically, an n-dimensional manifold is called *developable* if there exists an isometry between geodesic distances on the manifold and Euclidean distances in a convex subset of $\mathbb{R}^n$; intuitively, if the manifold can be unfolded without any distortions, it is developable. When confronted with non-developable manifolds, ISOMAP — both as a dimension reduction and intrinsic dimension estimation method — does not yield satisfying results

in many cases. Moreover, the approach can be highly sensitive when it comes to the proper choice of the particular graph parameter, be it k or ϵ.

The LLE method (see [RS00, RS03]) is motivated by the MDS concept, however, instead of a global optimization process including all pairwise distances, it seeks to preserve local similarities between neighboring points. For this purpose, in a first step, each data point $\boldsymbol{x}_i \in \mathbb{R}^D$ is reconstructed via a weighted combination of its nearest neighbors. More precisely, the cost function

$$f_{\mathrm{LLE}}^{(1)}(W) = \sum_{i=1}^{N} \Big\| \boldsymbol{x}_i - \sum_{j=1}^{N} w_{ij} \boldsymbol{x}_j \Big\|^2 \tag{3.27}$$

is minimized, where the weight matrix $W = [w_{ij}]$ is required to fulfill two constraints: it is sparse in the sense that, for each $i = 1, \ldots, N$, $w_{ij} \neq 0$ only if $\boldsymbol{x}_j$ belongs to the k nearest neighbors of $\boldsymbol{x}_i$. Second, each row sums up to one, i.e., $\sum_j w_{ij} = 1$ for each i. This minimization problem can be solved using constrained least squares fits, which involve the solution of local linear systems with an associated Gram matrix. In certain cases, for example when $k > D$, this Gram matrix can be nearly singular and therefore must be conditioned using a small regularization parameter $\delta > 0$.

In the second step, for fixed output dimension m, LLE computes a lower-dimensional representation $\hat{\boldsymbol{x}}_1, \ldots, \hat{\boldsymbol{x}}_N \in \mathbb{R}^m$ of the data by minimizing the cost function

$$f_{\mathrm{LLE}}^{(2)}(\hat{X}) = \sum_{i=1}^{N} \Big\| \hat{\boldsymbol{x}}_i - \sum_{j=1}^{N} w_{ij} \hat{\boldsymbol{x}}_j \Big\|^2, \tag{3.28}$$

where the minimization is now carried out with respect to the matrix $\hat{X} \in \mathbb{R}^{N \times m}$, while the weights w_{ij} are the outcome of step one. In order to get a well-posed minimization problem, the authors introduce the following two constraints: the outputs are required to be centered at the origin, i.e., $\sum_{i=1}^{N} \hat{\boldsymbol{x}}_i = \boldsymbol{0}$, and to have unit covariance, i.e., $\frac{1}{N} \sum_{i=1}^{N} \hat{\boldsymbol{x}}_i \hat{\boldsymbol{x}}_i^T = \boldsymbol{I}_m$. The solution can be found via computing the smallest $m+1$ eigenvalues and corresponding eigenvectors of the symmetric and positive-semidefinite matrix $A = (\boldsymbol{I}_N - W)^T (\boldsymbol{I}_N - W)$. Increasing or decreasing the number of eigenvalues yields hierarchically embedded solutions of higher or lower dimension m, respectively. Theoretically, the jumps between the eigenvalues (arranged in ascending order) can also be used for estimating the intrinsic dimensionality of the data.

Due to its local nature, LLE is able to successfully recover certain nonlinear manifolds. This capability actually stems from its first step: the computation of nearest neighbors and the associated restriction of the weights. However, in [LV07], Lee and Verleysen remark that different choices of the two parameters, the number k of NNs and the regularization constant δ, can lead to highly divergent results.

When it comes to the estimation of the intrinsic dimension, the authors of [PP02] examine some preconditions on the data that allow for the use of LLE as an IDE method. On the other side, the inventors of the method recommend against this usage in [RS03]

and present some rather simple examples of two-dimensional manifolds, where the eigenvalue spectrum in the last step of LLE can not provide an unambiguous criterion.

In summary, it must be noted that the majority of projective methods have been derived for the purpose of dimensionality reduction and thus, their usefulness in dimension estimation — even though being a convenient side benefit — is often limited.

Fractal methods

Under the notion of *fractal methods* we shall summarize all approaches which are based directly on one of the concepts of fractal dimensions introduced in section 3.1. Most of these methods rely on the "manifold model": the given data points are assumed to be sampled through some smooth probability density function from an underlying manifold with intrinsic topological dimension m. Most often, the theoretical concept is in fact based on the premise of uniformly distributed data.

According to [CCCR15], only two state-of-the-art estimators seek to explicitly estimate the Lebesgue covering dimension. The first is the so-called *tensor voting framework* (see [MMN05, MM10] for details) and the second approach in [LGX09] is motivated by LLE and estimates the ID using local weights that approximate each point by its nearest neighbors, in a very similar manner as in the first step of LLE. As remarked both in [CCCR15] and the original publications, those two methods suffer from certain drawbacks, especially when it comes to data with higher ID values.

In contrast, approaches based on estimating specific fractal dimensions, such as the box-counting or correlation dimension, are widespread and have been successfully applied in many areas. The most famous of all those methods is the estimator of the correlation dimension (see subsection 3.1.4) introduced by Grassberger and Procaccia in [GP83]. This variant will be included in our numerical experiments and is described in detail in subsection 3.2.2. It should be noted here that, in theory, this particular method requires a prohibitively large number of data points as shown in [Smi88] and [ER92]. To be specific, the ID m and the number of points N must satisfy

$$m < 2 \log_{10} N. \tag{3.29}$$

Nevertheless, the Grassberger-Procaccia estimator can still be an adequate choice for small values of m, and heuristic correction methods, e.g. as proposed in [CV02], can effectively attenuate its deficiencies.

Another important variant using a maximum likelihood rule to yield an asymptotically unbiased estimator of the correlation dimension has been proposed by Takens in [Tak85]. After choosing a threshold value ϵ_1, only pairs of points with distance less than ϵ_1 are considered. Let $r_1, \ldots, r_K \in [0, 1]$ denote those distances divided by ϵ_1. Now, it can be shown that

$$\hat{m} = -\left(\frac{\sum_{i=1}^{K} \ln(r_i)}{K} \right)^{-1} \tag{3.30}$$

is an assymptotically unbiased estimator for the correlation dimension with assymptotically minimal variance.

The Grassberger-Procaccia and the Takens estimators are analyzed and compared in [The90] and [BBD99]. In the latter publication, the authors suggest to modify the Takens estimator by introducing a lower threshold value ϵ_0 and evaluating only pairwise distances within the range $[\epsilon_0, \epsilon_1]$. In practice, estimation results will be highly dependent on the choice of those two parameters.

A promising approach presented in [HA05] is based on a generalization of the correlation sum (eq. (3.9)) in the setting of the correlation dimension. The Heaviside step function is replaced by a universal kernel function. Besides, an automatic scheme for selecting multiple scales is introduced, which only leaves a smallest scale to choose. Finally, a convergence analysis based on so-called U-statistics is provided and a comparison with the estimators by Grassberger-Procaccia and Takens proves the effectiveness of the new method. A precise description is given in subsection 3.2.2, and the algorithm is also included in our numerical experiments in section 3.5.

Numerous further methods exploiting the concept of fractal dimensions exist — we only mention a few more important variants. Ashkenazy's approach [Ash99] relies on the general information dimension, while Kégl [Kég03] uses packing numbers to approximate the capacity dimension, a setup that is generalized in [RL06]. A method optimized for binary datasets is presented in [TMGM06].

Nearest-neighbor based methods

Many IDE methods analyze certain properties of local sets of nearest neighbors (NN) to derive an estimate of the intrinsic dimension. Starting with the model of the underlying m-dimensional manifold, most of these methods act on the assumption that, given some data point $\boldsymbol{x}_i$ and a sufficiently small radius r, the corresponding points within the m-dimensional ball $B_r(\boldsymbol{x}_i)$ around $\boldsymbol{x}_i$ with radius r are distributed according to the multivariate uniform (or some other model-specific) distribution. Afterwards, certain quantities e.g. like the number of points in a particular region, the distribution of distances or angles are evaluated for each nearest neighbor subset to yield local estimates.

The first steps in NN based estimators have been made by Trunk in [Tru76] and by Pettis et al. in [BDJP79]. In a nutshell, Trunk's method fixes a number k of nearest neighbors and, for a given data point $\boldsymbol{x}_i$, considers the angle θ between the $(k+1)$th nearest neighbor of $\boldsymbol{x}_i$ and the subspace spanned by the k NNs of $\boldsymbol{x}_i$. The average over those angles (for all data points) is then used to get a proper ID estimate. However, the choice of a proper threshold parameter is nontrivial and moreover, the method has only been tested successfully for low ID values.

The approach by Pettis et al. assumes that the N data points are drawn independently according to some unknown density $p(\cdot)$. Their starting point is the following density estimator

$$\widehat{p}(\boldsymbol{x}) = \frac{k}{N} \cdot (V_m r_k^m)^{-1}, \tag{3.31}$$

where k represents the number of considered NN points, r_k is the distance of $\boldsymbol{x}$ to its kth nearest neighbor, and V_m is the volume of the m-dimensional unit ball. Then, an iterative scheme is developed to estimate the intrinsic dimension m from a plot of $\log(\overline{r}_k)$ versus $\log(k)$ for $k = 1, \ldots, k_{\max}$, where $\overline{r}_k$ denotes the global average distance to the kth nearest neighbor. In the original publication [BDJP79], the method is evaluated only for datasets of low ID $m \leq 3$. Subsequent experiments performed in [VD95] however led to the conclusion that this estimator tends to underestimation, especially for datasets of moderate or high intrinsic dimensionality.

Another interesting IDE method proposed in [FQZ09] is also based on eq. (3.31). It relies on counting the number of nearest neighbors in balls of growing radii, and fits a polynomial to those empirical values. In the end, the degree of such a polynomial fulfilling several constraints is considered the final ID estimate. The algorithm requires the selection of multiple parameters, however, it showed a competitive performance when compared to the well-established maximum likelihood estimator [BL05], which we discuss next.

This estimator (abbreviated MLE) introduced by Bickel and Levina has quickly become very popular in many application areas. Its model is based on a Poisson process which describes the number of neighbors within distance t as a time-dependent process. The final point-wise estimate is derived via a maximum likelihood approach. The associated formula is quite simple and depends on nothing but the distances to the k nearest neighbors. A detailed explanation can again be found in our comparison in subsection 3.2.2. Consistency of this method (and similar ones) has been proven in [PY13]. Also, further analysis in [MG05] seeks to address the weakness of the MLE's high bias. Finally, multiple extensions of the original method have been presented, which we will also discuss below.

Next, a group of researchers including Campadelli et al. have presented three similar approaches for intrinsic dimension estimation in [CCC+11], [CCL+11] and [CCC+12], as well as [CCB+14]. Apart from that, the same authors also suggested the benchmark framework in [CCCR15].

The first algorithm called "MiND$_{ML}$" (see [CCC+11]) is a maximum likelihood approach similar to the estimator in [BL05]. Given the d-dimensional unit ball $B_1(\mathbf{0})$ and k points $\boldsymbol{z}_i$ uniformly drawn from its interior, the authors consider the probability density function $g(r; d, k)$ related to the event $r = \min_{i \in \{1,\ldots,k\}} \|\boldsymbol{z}_i\|$. Given a fixed value of nearest neighbors k, a maximization of the log-likelihood function for all values $d = 1, \ldots, D$ yields the desired local estimate. As reported in [CCC+12], this method still suffers from serious underestimation for high ID values.

The second approach, presented in [CCL+11] and revisited in [CCC+12], is named "IDEA" and based on the fact that uniform sampling within a d-ball is equivalent to sampling from a multivariate standard normal distribution with subsequent scaling. The

crucial observation is that the quantities

$$f(\boldsymbol{x}_i) = 1 - \frac{T_j(\boldsymbol{x}_i)}{T_{k+1}(\boldsymbol{x}_i)} \tag{3.32}$$

are distributed according to the beta distribution $\beta_{1,d}$. Here, $T_j(\boldsymbol{x}_i)$ denotes the distance of $\boldsymbol{x}_i$ to its jth nearest neighbor among all given data points. This leads to the following compact estimator for the intrinsic dimension m:

$$\hat{m} = \frac{\hat{d}}{1-\hat{d}}, \quad \text{where} \quad \hat{d} = \frac{1}{Nk} \sum_{i=1}^{N} \sum_{j=1}^{k} \frac{T_j(\boldsymbol{x}_i)}{T_{k+1}(\boldsymbol{x}_i)}. \tag{3.33}$$

Since this approach also suffers from underestimation for high IDs, the authors implement an asymptotic correction step as proposed in [CV02], which basically consists in applying the original estimator to random subsets of different sizes (and associated values of k, the number of NNs), and calculating an asymptotic value from those different estimates based on empirical observations. This procedure comes at the expense of a higher complexity, however, according to [CCC+12], it lessens the negative effects of underestimation.

Finally, in [CCB+14], a third method called "DANCo" is introduced. It seizes on the basic concept of the MiND_{ML} approach, which is the probability density function (PDF) modeling the distribution of nearest neighbor distances, and adds a second PDF modeling the distribution of pairwise angles. The name refers to the concept of *dimensionality estimation via angle and norm concentration*. In a nutshell, the method compares the statistics of the two particular PDFs with those pre-computed on synthetic data of known ID, which is accomplished via Kullback-Leibler-divergences. Since this approach has been tested with promising results in [CCCR15], it is included in our upcoming comparison and the exact details as well as numerical results can be found below.

Many more methods based on nearest neighbor distances exist. Eventually, we would like to mention the approach by Audibert et al. in [FSA07], which leads to another simple formula for a local estimate at the point $\boldsymbol{x}_i$:

$$\hat{m}(\boldsymbol{x}_i) = \frac{\ln 2}{\ln\left(T_k(\boldsymbol{x}_i)/T_{\lceil k/2 \rceil}(\boldsymbol{x}_i)\right)}. \tag{3.34}$$

The global ID estimate is then computed either via averaging or voting, i.e., selecting the value $\hat{m}$ that appears most often among the local results. Furthermore, the authors prove the consistency (in probability) and provide an in-depth analysis of the reliability of their estimator under certain regularity assumptions on the underlying manifold. These theoretical bounds must however be considered with care, since first, they include three universal constants and second, even though the calculated bounds suggest that voting should perform better than averaging, the experimental results carried out by the authors show a precisely inverse behavior.

Altogether, a comprehensive comparison of existing NN methods would certainly be very informative, however — to our knowledge — it has not been realized yet.

Simplex based methods

Next, we summarize IDE techniques based on the examination of simplices. Since our method is based on the evaluation of simplex volumes, we would like to clearly distinguish it from similar approaches.

In the context of statistical shape analysis (see [DM98, SG02]), a particular type of polytopes has been used in dimension estimation. In short, statistical shape analysis is the examination of geometric properties of one or several shapes, given by either an exact representation or a point sampling. This research area has been motivated mainly by biological and medical problems. Thus, the shapes are often derived from certain physical, three-dimensional objects, such as bones or blood vessels.

In order to map test points from one object onto another similar object, the question of determining the local dimension (usually ranging in the interval $[1, 3]$) of data points sampled from an underlying continuous object has emerged and has been investigated e.g. in [DGGZ03, GW04]. First, the data points $\boldsymbol{x}_{i=1,\dots,N} \in \mathbb{R}^D$ are assumed to be a so-called (ϵ, δ)-*sampling* of the underlying manifold, which ensures that the sampling is sufficiently dense where required, but at the same time does not form arbitrary low-dimensional patterns disturbing the estimation procedure.

Next, the *Voronoi decomposition* with respect to the data points is considered. This is a partition of the space, where each $\boldsymbol{x} \in \mathbb{R}^d$ is assigned to the point $\boldsymbol{x}_i$ with minimum Euclidean distance. Each $\boldsymbol{x}_i$ is now called the *center* of its *Voronoi cell* or *polytope* $V_{\boldsymbol{x}_i}$, i.e., the set of all points assigned to $\boldsymbol{x}_i$. Now, the authors remark that $V_{\boldsymbol{x}_i}$ approximates the Minkowski sum of $T_{\boldsymbol{x}_i} \cap V_{\boldsymbol{x}_i}$ and $N_{\boldsymbol{x}_i} \cap V_{\boldsymbol{x}_i}$, where $T_{\boldsymbol{x}_i}$ and $N_{\boldsymbol{x}_i}$ denote the tangent space and the normal space, respectively, at $\boldsymbol{x}_i$ with respect to the manifold. Finally, the heights of certain lower-dimensional subpolytopes of $V_{\boldsymbol{x}_i}$ are evaluated in order to determine local ID estimates.

This approach has been successfully applied to subdivide three-dimensional manifolds into local areas that are intrinsically either one-, two-, or three-dimensional. However, the algorithm has been tailored exactly for this specialized context, and therefore, its capabilities of dealing with higher-dimensional data are probably rather limited.

A method motivated by the former one has been presented in [CC09] and utilizes the notion of *sliver*, being a degenerated simplex with small "negligible" volume. The concept of slivers has emerged in the context of mesh generation, cf. [CFF85, CDE+00], and *Delaunay triangulations* (a triangulation dual to the Voronoi decomposition), where it is often beneficial to avoid or eliminate slivers in order to get structures fulfilling certain regularity constraints. There are at least two slightly different definitions of the term "sliver", which we introduce now. First, a parameter $\sigma \in (0, 1)$ is fixed, quantifying the degree of degeneration. Then, a σ-*sliver* is defined as follows.

(I) All 0- and 1-simplices are no σ-slivers.

(IIa) For $d \geq 2$, a d-simplex Δ is called σ-sliver, if none of its boundary simplices is a σ-sliver, and there exists one boundary simplex Δ_p, such that

$$\mathrm{vol}(\Delta) < \sigma \cdot l_\Delta \cdot \mathrm{vol}(\Delta_p), \tag{3.35}$$

where l_Δ denotes the length of the shortest edge of Δ.

(IIb) *Alternatively*: For $d \geq 2$, a d-simplex Δ is called σ-sliver, if

$$\mathrm{vol}(\Delta) < (\sigma \cdot L_\Delta)^d / d!, \tag{3.36}$$

where L_Δ denotes the length of the longest edge of Δ.

The first variant (IIa), presented e.g. in [CDR05], is more commonly used, while the second variant (IIb) is utilized in [CC09], where the authors claim that the differences between the two alternatives are minor under their specific assumptions on the point sampling.

The approach introduced in [CC09] is based on a theorem from [CDR05]. This theorem states that, for a given manifold of dimension m and an associated (ϵ, δ)-sample set S, under certain conditions on the parameters ϵ, δ, and σ, there exists a restricted weighted Delaunay triangulation of S, which is homeomorphic to the manifold and does not contain any j-dimensional σ-sliver for all $1 \leq j \leq m$. Furthermore, any $(m+1)$-dimensional simplex with vertices selected from neighboring points of S is a σ-sliver, provided that none of its boundary simplices is a σ-sliver. For the complete details, we refer to the original publication.

This theorem can now be exploited in order to estimate the dimension via analyzing multiple j-simplices made up of neighboring points of S for growing values of $j \geq 2$. The number of slivers found for the current test dimension j is then used as the decision rule for either $j \leq \hat{m}$ or $j > \hat{m}$, where $\hat{m}$ is the estimated intrinsic dimension.

The practical implementation of the method uses three different parameter combinations to distinguish between sliver and non-sliver simplices and it thus appears strongly dependent on a proper choice of parameters. It has been tested only for d-dimensional spheres (with $4 \leq d \leq 9$) and two different, intrinsically low-dimensional real datasets, both with solid results.

Finally, a promising IDE approach can be found in [JSF15], where the *skewness* of simplices made up of neighboring points is utilized to estimate the intrinsic dimension. The method is called "expected simplex skewness" and it exploits a general concentration phenomenon. A detailed description and numerical results are provided in subsection 3.2.2 and section 3.5, respectively.

Graph based methods

A thorough overview on current graph based IDE approaches is provided by Brito et al. in [BQY13]. Also, an empirical comparison between several different methods is included.

According to the authors, graph based approaches benefit from lower computational costs, the comprehensive availability of theory, and a certain robustness against noise, since most graph constructions rather rely on the ordering of nearest neighbor distances than on the exact distance values. They revisit their IDE method presented in [BQY02] and introduce two new similar methods based on different graph concepts.

The three particular graphs are the well-known *k-nearest-neighbor graph* $\mathbf{G}_k$, the *minimum spanning tree graph* $\mathbf{M}$, and the *sphere of influence graph* $\mathbf{S}_k$. In $\mathbf{G}_k$, two points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are connected if and only if either $\boldsymbol{x}_j$ is within the set of the k NNs of $\boldsymbol{x}_i$ or vice versa.

The minimum spanning tree (MST) graph $\mathbf{M}$ is a spanning tree with minimal total weight; here, an edge weight corresponds to the Euclidean distance between the two vertex points, and a spanning tree denotes an undirected graph connecting all given points. Replacing the Euclidean distance by the geodesic distance yields the *geodesic minimum spanning tree* (GMST).

In the sphere of influence graph $\mathbf{S}_k$, two points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are connected if and only if $\|\boldsymbol{x}_i - \boldsymbol{x}_j\| \leq T_k(\boldsymbol{x}_i) + T_k(\boldsymbol{x}_j)$, where $T_k(\boldsymbol{x}_i)$ denotes the distance of $\boldsymbol{x}_i$ to its kth nearest neighbor point.

The first approach introduces the notion of *reach* $r_t(\boldsymbol{x}_i, \mathbf{G}_k)$, which equals the number of points $\boldsymbol{x}_j$ that can be reached from $\boldsymbol{x}_i$ in the graph $\mathbf{G}_k$ via a path $\boldsymbol{x}_i = v_0, v_1, \ldots, v_t = \boldsymbol{x}_j$ within t steps or less. The intuition is that the statistic about the average reach of t steps,

$$\bar{r}_t(\mathcal{X}, \mathbf{G}_k) = \frac{1}{N} \sum_{i=1}^{N} r_t(\boldsymbol{x}_i, \mathbf{G}_k), \tag{3.37}$$

depends on the intrinsic dimension m, since in higher dimensions, a point has more directions to connect to neighbors and thus, the sum in eq. (3.37) should increase with the value of m.

The second approach based on the MST graph $\mathbf{M}$ considers the quantity

$$\bar{d}(\mathcal{X}, \mathbf{M}) = \frac{1}{N} \sum_{i=1}^{N} (\deg(\boldsymbol{x}_i))^2, \tag{3.38}$$

where the *degree* $\deg(\boldsymbol{x}_i)$ of node $\boldsymbol{x}_i$ equals the number of points $\boldsymbol{x}_j$ such that $\{\boldsymbol{x}_i, \boldsymbol{x}_j\}$ is an edge of $\mathbf{M}$. It has been shown in [SSE87] that, given N samples of a continuous distribution, the number of graph nodes with a fixed degree t scales with the intrinsic dimension m in the limit $N \to \infty$ almost surely. This justifies the use of the above statistic.

For the sphere of influence graph $\mathbf{S}_k$, the authors count the number $n_{i,j}$ of points, excluding $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, in the intersection $B_{T_k(\boldsymbol{x}_i)}(\boldsymbol{x}_i) \cap B_{T_k(\boldsymbol{x}_j)}(\boldsymbol{x}_j)$, i.e., the number of points among the k nearest neighbors that $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ share. The corresponding statistic is then defined as

$$\bar{u}(\mathcal{X}, \mathbf{S}_k) = \frac{1}{N} \sum_{1 \leq i < j \leq N} n_{i,j}. \tag{3.39}$$

For each of those three statistics, to yield a valid estimator, it is assumed that the underlying probability density corresponds to a Gaussian density with mean $\mu(m)$ and variance $\sigma^2(m)/N$. Here, the two latter parameters are supposed to depend on the intrinsic dimension m only. Finally, a Bayesian decision theoretic approach is applied to derive an a posteriori expected value of the intrinsic dimension, where the reference values are pre-computed in a simulation of random samples of the uniform distribution on the d-dimensional unit cube.

After a recommendation for the proper selection of the required parameters for each statistic, a comparison with other methods, including the MLE method [BL05], for various datasets (sphere, paraboloid, and swiss roll) is provided in [BQY13]. While the graph based methods feature competitive results, the authors still come to the conclusion that none of the methods is clearly superior when compared to the others.

Two different graph based IDE algorithms included in the comparison are due to Hero et al. (see [CHG05] and [SRH10]). The earlier approach has already been described in [CH04b, CH04a] and is based on the following length functional

$$L_\gamma(\mathcal{X}, \mathbf{G}) = \sum_{e \in \mathbf{G}} |e|^\gamma, \tag{3.40}$$

where $\mathbf{G}$ is a so-called *entropic graph* (e.g. the k-nearest-neighbor graph or minimum spanning tree graph) and e is an edge in $\mathbf{G}$. Further, $\gamma \in (0, d)$ is a weight parameter which will ultimately be used for the dimension estimation. The authors exploit a theorem from [BHH59] which states that, under certain assumptions on the distribution (with density function f) of the sample points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N \in \mathbb{R}^d$, the limit converges with probability 1 as

$$\lim_{N \to \infty} L_\gamma(\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}, \mathbf{G})/N^\alpha = \beta_d \int_{\mathbb{R}^d} f^\alpha(\boldsymbol{y}) d\boldsymbol{y}, \tag{3.41}$$

where $\alpha = (d - \gamma)/d$ and β_d is a constant not depending on the underlying distribution. The same asymptotic behavior can now be shown for points sampled from an underlying manifold with intrinsic dimension m, where d is replaced by m in eq. (3.41). Finally, a non-parametric least squares strategy is applied to estimate m from multiple trials with different subset sizes.

In [CRH07], the IDE method from [CHG05] is slightly modified in two ways. First, it is converted into a local estimator. Second, the authors suggest a general approach for de-biasing such estimators in order to avoid the dominating influence of the boundary effect in high dimensions. Their concept of a measure for the "data depth" of points looks promising, however, further examinations in [CRH10] showed that for increasing intrinsic dimension m, the theoretical potential of improvement vanishes.

Furthermore, Hero et al. propose another IDE variant in [SRH10] based on the k-nearest-neighbor graph $\mathbf{G}_k$. In fact, this approach does not require specific properties of the graph, but only relies on NN distances, and should thus rather be classified as a

nearest-neighbor based method. The idea is to partition the dataset into two sets, and consider the average of the logarithms of k-NN distances of all points of the first set to the neighbors of the second set. More precisely, this statistic is evaluated two times for two different values of $k = k_1, k_2$, and the comparison of the outcomes allows for the estimation of the intrinsic dimension.

As mentioned before, both procedures introduced in [CHG05] and [SRH10] have been examined in [BQY13] amongst other up-to-date graph based methods with no definite winner.

A very recent examination of geodesic distances combined with the scale-dependent correlation dimension is presented in [GC16]. Here, the authors analyze similarities of log-log-plots of the scale-dependent correlation dimension (with Euclidean distances replaced by geodesic ones) of different sampled objects of the same dimensionality in order to estimate the ID for other datasets. The general concept of this approach (with Euclidean distances) will be explained in detail in the upcoming subsection. In [GC16], some promising results are achieved for simple geometric objects of IDs up to $m = 20$ and also for certain low-dimensional, but more complex structures such as the swiss roll and the 10-fold Möbius strip.

Summary

As can be seen from the above considerations, many different approaches for the problem of estimating the intrinsic dimension of a dataset exist. We introduced the — to the best of our knowledge — most relevant methods based on projections, fractal dimensions, the analysis of nearest neighbors, simplices, and graphs, where an unambiguous classification is not always possible. Due to the vast number of IDE procedures, originating from diverse research areas, it is beyond the scope of our work to cite *all* existing methods. As mentioned before, we selected six particular approaches that will be described in detail in the following subsection and will be part of the subsequent numerical comparison of section 3.5.

3.2.2 Selected approaches

In the following, we present the selected methods that are significant in our context for different reasons. The classic estimator of the correlation dimension by Grassberger and Procaccia [GP83] is chosen as our reference method, since it is one of the most cited and well-established IDE methods.

We continue with a closely related approach by Hein and Audibert [HA05] that we choose to call "generalized correlation dimension", since it basically generalizes and extends Grassberger and Procaccia's concept. Moreover, in [HA05], the authors suggest a number of synthetic datasets for the study of key properties of IDE methods, that we will come back to in our own numerical experiments later on.

Next, we focus on the *maximum likelihood approach* (MLE) by Bickel and Levina [BL05] that is well-understood and has led to several refined approaches, e.g. in [HRS08], where noise is included in the statistical model, further [DGH10], where a regularization term is added to the original model, and [KDM11, KD15], where geodesic distances are used instead of Euclidean distances. The MLE method has repeatedly been used in miscellaneous applications (see [LWC+07, SWLH10]), also for the purpose of selecting the target dimension for dimension reduction techniques, e.g. in [vdMPvdH09], [OV13].

Less attention has been paid so far to an estimator introduced by Chávez et al. in [CNBYM01], that relies on nothing but the empirical mean and variance of pairwise distances and is simply referred to by the authors as "intrinsic dimensionality" of a metric space. We did not mention this variant before since it is not a concrete procedure, but rather a theoretical concept, which, however, has also been promoted in an axiomatic analysis by Pestov in [Pes08].

A recent approach presented in [CCB+14] by Campadelli et al. called "Dimensionality from Angle and Norm Concentration" (DANCo) is particularly interesting for us from a theoretical viewpoint since it exploits concentration effects of both distances and angles. In practical experiments, e.g. in [CCCR15], it turned out to be relatively robust both for data tainted with moderate noise and for data of high intrinsic dimensionality.

The method introduced in [JSF15] by Johnsson et al. is noteworthy for several reasons. First, it relies on the analysis of the expected simplex skewness (ESS), which is a dimension dependent expected value, and thus, it is similar to our approach. Second, according to the authors, it delivers competitive results, however, to our knowledge, it has not been tested elsewhere. Beyond that, ESS is able to provide quite reliable estimates of the intrinsic dimension for the case $m > N$. This is an unusual feature of an IDE method and motivates further considerations.

For each approach described below, we proceed in the following way: we start with the description of the model and the assumptions about the input data. Then, the particular estimator and the role of its parameters are explained. Next, we highlight the most important characteristics and finally, if applicable, we point out similar methods and known extensions.

Correlation Dimension (CD)

The concept of the correlation dimension ([GP83]) has already been introduced in subsection 3.1.4. Given a finite set of data $\mathcal{X} = \{\boldsymbol{x}_{i=1,\dots,N}\}$, the principal idea is to consider the total number of pairs of points with a Euclidean distance of less than a given $\epsilon > 0$, and then analyze those values for appropriately small $\epsilon \to 0$. More precisely, the correlation sum for finitely many points is defined by

$$C_\epsilon(\mathcal{X}) = \frac{2}{N(N-1)} \sum_{1 \le i < j \le N} H\left(\epsilon - \|\boldsymbol{x}_i - \boldsymbol{x}_j\|\right). \tag{3.42}$$

The estimated correlation dimension is then given by

$$\hat{m}_{\text{cor}} = \lim_{\epsilon \to 0} \frac{\log C_\epsilon(\mathcal{X})}{\log \epsilon}. \tag{3.43}$$

Since both the numerator and the denominator tend to $-\infty$ for $\epsilon \to 0$, the above definition is not directly applicable in practice. However, applying de l'Hospital's rule yields

$$\begin{aligned} \hat{m}_{\text{cor}} &= \lim_{\epsilon \to 0} \frac{\partial_\epsilon \log C_\epsilon(\mathcal{X})}{\partial_\epsilon \log \epsilon} \\ &= \lim_{\epsilon_2, \epsilon_1 \to 0} \frac{\log C_{\epsilon_2}(\mathcal{X}) - \log C_{\epsilon_1}(\mathcal{X})}{\log \epsilon_2 - \log \epsilon_1} \\ &= \lim_{\epsilon_2, \epsilon_1 \to 0} \frac{\log \frac{C_{\epsilon_2}(\mathcal{X})}{C_{\epsilon_1}(\mathcal{X})}}{\log \frac{\epsilon_2}{\epsilon_1}}. \end{aligned} \tag{3.44}$$

To get a viable estimator, we still must determine a valid range of values for ϵ_i and furthermore, we have to practically evaluate the limit towards zero. For this purpose, we follow the suggestions in [LV07]. To begin with, it is obvious that neither too large values of ϵ_i are a good choice, since we are considering the limit of $\epsilon_i \to 0$, nor too small values, since $C_\epsilon(\mathcal{X})$ will be zero. For this reason, in a first step, the global minimum and maximum of all pairwise distances, $\delta_{\min}$ and $\delta_{\max}$, are computed. For a constant number L, the range $[\delta_{\min}, \delta_{\max}]$ is now divided into L intervals uniformly stretched across the logarithmic scale via

$$\epsilon_i = \delta_{\min} \cdot \left(\frac{\delta_{\max}}{\delta_{\min}} \right)^{i/L}, \qquad i = 0, \ldots, L. \tag{3.45}$$

For two neighboring values ϵ_i and ϵ_{i+1}, the following scale-dependent correlation dimension estimate is then given as

$$\hat{m}_{\text{cor}}^{(i)} := \hat{m}_{\text{cor}}\left(\epsilon_{i+1}, \epsilon_i\right) = \frac{\log \frac{C_{\epsilon_{i+1}}(\mathcal{X})}{C_{\epsilon_i}(\mathcal{X})}}{\log \frac{\epsilon_{i+1}}{\epsilon_i}}. \tag{3.46}$$

In [LV07], the authors recommend to find the largest "plateau" in the corresponding log-log plot, i.e., the largest region with almost constant slope, in order to get a good estimate. We implement this recommendation in the following way. In a first step, we determine the index

$$q = \operatorname*{arg\,min}_{i=2,\ldots,L-3} \left(\frac{\sum_{j=i-2}^{i+2} \left| \hat{m}_{\text{cor}}^{(i)} - \hat{m}_{\text{cor}}^{(j)} \right|}{\hat{m}_{\text{cor}}^{(i)}} \right). \tag{3.47}$$

The idea is to extract the local value $\hat{m}_{\text{cor}}^{(q)}$ with minimum average (relative) distance to

its neighboring estimates. The final result is then defined as the average of these five contiguous values:

$$\hat{m} = \frac{1}{5} \sum_{j=q-2}^{q+2} \hat{m}_{\text{cor}}^{(j)}. \tag{3.48}$$

For this approach, we found that $L = 20$ is a reasonable and sufficient number of intervals.

Generalized Correlation Dimension (GCD)

The setting of the approach proposed in [HA05] by Hein and Audibert is as follows. The data $\mathcal{X} = \{\boldsymbol{x}_{i=1,\dots,N}\} \subset \mathbb{R}^D$ are samples of a probability measure P with support on an m-dimensional submanifold $M \subset R^D$. The submanifold is required to satisfy certain regularity assumptions. These can be essentially subsumed by smoothness (M has to be a *Riemannian manifold*), being well-behaved (which includes bounded *extrinsic* and bounded *sectional curvature*), and global boundedness of the *injectivity radius* and the *regularity radius* away from the boundary of M. The precise definitions from the field of Riemannian geometry can either be found in the original paper [HA05] or in [Lee97].

Next, the probability density function $f : M \to \mathbb{R}_+$ of P is defined with respect to the natural volume element $dvol(\boldsymbol{x})$ on M, and furthermore, it is assumed to be three times differentiable and square-integrable.

The starting point of the dimension estimation approach is the correlation sum for finitely many points, defined in subsection 3.1.4, associated with the correlation dimension:

$$C_\epsilon(\mathcal{X}) = \frac{2}{N(N-1)} \sum_{1 \le i < j \le N} H\left(\epsilon - \|\boldsymbol{x}_i - \boldsymbol{x}_j\|\right). \tag{3.49}$$

The main idea is to replace the Heaviside function in the correlation sum by a general isotropic kernel function $K : \mathbb{R}_+ \to \mathbb{R}$. The outcome is a so-called U-statistic[1] (we will explain this term in the following) of K:

$$U_{N,h,m}(K) = \frac{2}{N(N-1)} \sum_{1 \le i < j \le N} K_{h,m}\left(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2\right), \tag{3.50}$$

where $K_{h,m}$ is the corresponding m-dimensional kernel scaled with bandwidth h:

$$K_{h,m}\left(\|\boldsymbol{x} - \boldsymbol{y}\|^2\right) = \frac{1}{h^m} K\left(\|\boldsymbol{x} - \boldsymbol{y}\|^2 / h^2\right). \tag{3.51}$$

The kernel function K is required to fulfill certain constraints, in particular it must be measurable, non-negative, non-increasing, and its second derivative must be bounded. Additionally, it must be square-integrable with respect to the m-dimensional Lebesgue-

[1] For a proper use of the term *U-statistic*, the kernel function should rather be defined as a function of two variables: $K : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$. Since we consider isotropic kernels only, we prefer the simpler, self-explanatory notation used above.

measure: $\int_{\mathbb{R}^m} K(\|\boldsymbol{y}\|^2)d\boldsymbol{y} = c_1 < \infty$.

The term U-statistic is due to HOEFFDING (see [Hoe48]) and can be viewed as a generalization of the sample mean. Given a set of *i.i.d.* samples $z_1, \ldots, z_n \in \mathbb{R}$ of the probability density function f, and some measurable function $\phi : \mathbb{R}^s \to \mathbb{R}$ with $s \leq n$, the U-statistic associated with ϕ is defined as

$$U_n(\phi) = \frac{(n-s)!}{n!} \sum_{(i_1,\ldots,i_s)\in\Pi(n,s)} \phi\left(z_{i_1}, \ldots, z_{i_s}\right), \tag{3.52}$$

where $\Pi(n, s)$ denotes the set of all permutations $(i_1, \ldots, i_s)$ of size s chosen from $(1, \ldots, n)$. Now, if the expected value $\mu(f) = \mathbb{E}_f\left[\phi(z_1, \ldots, z_s)\right]$ exists, $U_n(\phi)$ is obviously an unbiased estimate of $\mu(f)$. Under certain assumptions, the U-statistic is also the optimal unbiased estimate in terms of having a minimum variance. For further details, the reader is referred to [Lee90] and [KB94].

Now let us return to our particular problem. The main objective is to estimate the intrinsic dimension m from the behavior of (3.50). For this purpose, both limit values for $N \to \infty$ and $h \to 0$ of $U_{N,h,m}(K)$ and their corresponding expectations are studied. The procedure in [HA05] is the following.

First, it is shown that the limit for $h \to 0$ of the expectation exists:

$$\lim_{h\to 0} \mathbb{E}\left[U_{N,h,m}(K)\right] = c_1 \int_M f(\boldsymbol{x})^2 dvol(\boldsymbol{x}). \tag{3.53}$$

After a theorem by HOEFFDING is used to control the deviation of $U_{N,h,m}$ from its expectation, the authors deduce that applying the two limits $N \to \infty$ and $h \to 0$, under the assumption $N \cdot h^m \to \infty$, yields

$$\lim_{N\to\infty} \lim_{h\to 0} U_{N,h,m}(K) = c_1 \int_M f(\boldsymbol{x})^2 dvol(\boldsymbol{x}), \qquad \text{in probability.} \tag{3.54}$$

Finally, it can be shown that replacing m by some smaller or greater value l, the limit converges to zero or infinity, respectively. Formally, for $N \cdot h^l \to \infty$ we have

$$\lim_{N\to\infty} \lim_{h\to 0} U_{N,h,l}(K) = \begin{cases} 0, & \text{in probability} \quad \text{if } \ l < m, \\ \infty, & \qquad\qquad\quad \text{if } \ l > m. \end{cases} \tag{3.55}$$

Note the similarity of this property to definition (3.3) of the Hausdorff dimension.

We now reached the point, where we can explain the details of the estimation method. Given a dataset $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$, the central idea is to evaluate the U-statistic for different sub-samples of five sizes $n \in \{\lceil N/5\rceil, \lceil N/4\rceil, \lceil N/3\rceil, \lceil N/2\rceil, N\}$ and multiple values of $h = h_l(n)$, which must be chosen appropriately. The corresponding considerations on

convergence in [HA05] eventually result in the choice

$$h_l(n) = h_{\min} \cdot \left(\frac{N \log n}{n \log N}\right)^{1/l}, \qquad \text{where} \quad h_{\min} = \frac{1}{N} \sum_{i=1}^{N} T_1(\boldsymbol{x}_i). \tag{3.56}$$

The constant $h_{\min}$ is the average of all N distances between each point to its first nearest neighbor (recall that $T_j(\boldsymbol{x}_i)$ is the distance of $\boldsymbol{x}_i$ to its jth nearest neighbor). According to the authors, different kernel functions K only entail negligible variations in the outcome. In disagreement with the assumption that K should be two times continuously differentiable, the efficiently computable kernel

$$K(z) = (1 - z)_+ \tag{3.57}$$

is chosen. The whole estimation procedure can now be subsumed as follows. First, for each sub-sample size n_r, $r = 1, \ldots, 4$ (excluding $n_5 = N$), multiple disjoint subsets are chosen. Then the average (with respect to the subsets of the same size n_r) of the corresponding U-statistics $U_{n_r, h_l(n_r), l}(K)$ are computed for different test dimensions $l = 1, \ldots, l_{\max}$. In the last step, for each dimension l, a line is constructed through the five points $\big(\log h_l(n_r), \log U_{n_r, h_l(n_r), l}(K)\big)$ for $r = 1, \ldots, 5$, using weighted least squares with weights $w(r) = 1/r$. Finally, the absolute value of the slope is computed for each line, and the line with minimal absolute value determines the final dimension estimate $\hat{m} \in \{1, \ldots, l_{\max}\}$. This choice is reasonable due to the fact that, in theory, the corresponding slope is given by $(m - l) \log h_l(n)$ for $n \to \infty$ and $h \to 0$.

The authors do not state a reason for their particular choice of the number — let us call it $N_{\text{sub}} = 5$ — of different sub-sample sizes. Consequently, N_{sub} can be treated as a parameter of the method, even though it is suggested that its influence on the results is very limited.

The approach by Hein and Audibert, just like the MLE method, is based on the statistical distribution of distances between nearest neighbor points. While straightforward estimation of the correlation dimension requires the selection of multiple scales ϵ (see subsection 3.2.2), the GCD method only fixes a single minimum scale $h_{\min}$ and provides a scheme to select the larger scales appropriately. The computational costs are again dominated by the distance calculations, as long as N_{sub} is moderately small as proposed by the authors.

Finally, a comparison conducted in [HA05] with two established estimators, i.e., the correlation dimension estimator (CD) and the Takens estimator (see [Tak85]), both applied with carefully selected scales, shows that in numerous tested scenarios either one or the other estimator yields results comparable to those of the GCD method.

Maximum Likelihood Estimation (MLE)

The maximum likelihood estimator (MLE) presented in [BL05] is based on the following assumptions. Let $\boldsymbol{y}_i$ be points sampled *i.i.d.* from some smooth probability density function f on $\mathbb{R}^m$, where both f and m are unknown. The data $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N \in \mathbb{R}^D$ are then generated via a continuous and sufficiently smooth mapping $g : \mathbb{R}^m \to \mathbb{R}^D$, $\boldsymbol{y}_i \mapsto g(\boldsymbol{y}_i) = \boldsymbol{x}_i$. Beyond that, for a given point $\boldsymbol{x}$, the model requires the density $f(\boldsymbol{x})$ to be approximately constant in a small ball $B_r(\boldsymbol{x})$ with fixed radius r around $\boldsymbol{x}$. The data are then considered as observations of a certain inhomogeneous Poisson process. For this purpose, first, the function

$$S(t, \boldsymbol{x}) = \sum_{i=1}^{N} \mathbb{1}_{\{\boldsymbol{x}_i \in B_t(\boldsymbol{x})\}} \tag{3.58}$$

is introduced, that describes the number of occurrences within distance t from $\boldsymbol{x}$. Omitting the dependence of $\boldsymbol{x}$, this function can be approximated by an appropriate Poisson process $P(t)$ with rate

$$\lambda(t) = f(\boldsymbol{x}) \cdot V_m \cdot m \cdot t^{m-1}, \tag{3.59}$$

where V_m is the volume of the unit ball in $\mathbb{R}^m$. Finally, it can be shown that the log-likelihood of $P(t)$ is given by

$$L(m) = \int_0^r \log \lambda(t) dP(t) - \int_0^r \lambda(t) dt. \tag{3.60}$$

According to [BL05], if one considers the limit $N \to \infty$, a corresponding maximum likelihood estimator for m exists with probability 1, is unique, and can be derived as

$$\tilde{m}_r(\boldsymbol{x}) = \left(\frac{1}{S(r, \boldsymbol{x})} \sum_{j=1}^{S(r,\boldsymbol{x})} \log \frac{r}{T_j(\boldsymbol{x})} \right)^{-1}, \tag{3.61}$$

where $T_j(\boldsymbol{x})$ is defined as the Euclidean distance of $\boldsymbol{x}$ to its jth nearest neighbor.

It is also possible to fix the number of nearest neighbors k instead of the ball radius r, which yields the following asymptotically unbiased estimator:

$$\hat{m}_k(\boldsymbol{x}) = \left(\frac{1}{k-2} \sum_{j=1}^{k-1} \log \frac{T_k(\boldsymbol{x})}{T_j(\boldsymbol{x})} \right)^{-1}. \tag{3.62}$$

In [BL05], the authors only deploy the second variant for the definition of their estimator. It is obvious that the choice of k in (3.62) is crucial, since different values are likely to result in different estimates. This problem is attenuated simply by averaging estimates for multiple values $k = k_1, k_1 + 1, \ldots, k_2$. Thus, the final estimator for our

input data $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ is given by

$$\hat{m} = \frac{1}{k_2 - k_1 + 1} \sum_{k=k_1}^{k_2} \hat{m}_k, \quad \text{where} \quad \hat{m}_k = \frac{1}{N} \sum_{i=1}^{N} \hat{m}_k(\boldsymbol{x}_i). \tag{3.63}$$

Here, the authors note that a deliberate, data dependent choice of the two parameters k_1 and k_2 can reduce the bias. However, at the same time, the method is supposed to be stable for different values of k. Eventually, for the sake of reproducibility, standard values of $k_1 = 10$ and $k_2 = 20$ are proposed.

The MLE method is a local method and relies on statistical properties of distances between nearest neighbors to estimate the intrinsic dimension. The algorithmic costs are dominated by the computation of the k_2 nearest neighbors for each data point. Since the underlying model is based on the characteristics of sample points in a relatively small ball, the method generally requires a large number of data, especially in higher dimensions. Furthermore, for a small number k of nearest neighbors, the estimate suffers from a rather high bias.

In order to address this limitation, the authors of [MG05] replace the arithmetic mean in (3.63) by the harmonic mean, which yields the modified estimator

$$\hat{m}^{\text{mod}} = \frac{1}{k_2 - k_1 + 1} \sum_{k=k_1}^{k_2} \hat{m}_k^{\text{mod}}, \quad \text{where} \quad \hat{m}_k^{\text{mod}} = \left(\frac{1}{N} \sum_{i=1}^{N} (\hat{m}_k(\boldsymbol{x}_i))^{-1} \right)^{-1}. \tag{3.64}$$

Another extension of the MLE approach is presented in [DGH10], where, for the purpose of regularization, a Kullback-Leibler divergence (see [KL51]) between the rate parameters of the Poisson process is introduced. The main effect is again a reduction of the bias for a small number of nearest neighbors. A comparison of those three procedures in [DGH10] reveals subtle differences, but the numerical results are qualitatively very similar.

In [HRS08], the model by Levina and Bickel is first extended to a translated Poisson model including the presence of noise in the data. In the next step, mixtures of translated Poisson processes are considered in order to be able to handle datasets generated by multiple densities with different intrinsic dimensions. The resulting method is then applied in the context of stratification learning, which is the task of classifying points from datasets of the above-mentioned kind.

Finally, in [KDM11] the MLE method is also successfully applied with geodesic distances instead of Euclidean distances yielding superior results in certain scenarios, which is coherent, since geodesic distances often permit to capture some information about the local structure of the underlying manifold.

Distribution of Distances (DD)

The next approach we address here is based on an interesting concept of intrinsic dimensionality introduced in section 7 of [CNBYM01] by Chávez et al. in the context of the

analysis of proximity searching in high-dimensional metric spaces. A further discussion of this notion can also be found in [CN00].

Consider the metric space $(\mathcal{Y}, d_{\mathcal{Y}})$ and a finite subset $\mathcal{S} \subset \mathcal{Y}$. Then, the *intrinsic dimensionality* of $\mathcal{S}$ is defined as

$$\dim_{\text{dist}}(\mathcal{S}) = \mu^2/2\sigma^2, \tag{3.65}$$

where μ and σ^2 are the mean and variance of the histogram of distances of $\mathcal{S}$.

In the definition provided in [CNBYM01], the term *intrinsic dimensionality* is actually deployed for the metric space $\mathcal{Y}$ itself, which, however, leaves open the question of the construction of the histogram of distances.

In [Pes07, Pes08], Pestov provides some axiomatic background and compares this statistical quantity with similar concepts. Further considerations on the topic can also be found e.g. in [FWV07].

In practice, this definition can be applied to yield a global estimate of the ID of a dataset simply by analyzing all pairwise distances at once. When dealing with data points sampled from a manifold with high curvature, however, it is reasonable to suppose that a local estimator will produce much better results than a global one. For this reason, we propose the following local variant for a given dataset $\mathcal{X} = \{\boldsymbol{x}_{i=1,\dots,N}\}$:

$$\hat{m}_{\text{dist}} = \frac{1}{N}\sum_{i=1}^{N} \dim_{\text{dist}}\left(\left\{\boldsymbol{x}_i^{(j)} \,\middle|\, j = 0, \dots, k-1\right\}\right), \tag{3.66}$$

where $\boldsymbol{x}_i^{(j)}$ is the jth nearest neighbor of $\boldsymbol{x}_i$ (in particular $\boldsymbol{x}_i^{(0)} = \boldsymbol{x}_i$), and k is a fixed parameter.

This ansatz certainly is one of the most elementary estimators based on statistics of local distances. In addition, to our knowledge, the variant presented here has not been examined in practical experiments anywhere else, which is why we choose to include it in our subsequent comparison.

Angle and Norm Concentration (ANC)

Campadelli et al. introduced a method named DANCo (short for "Dimensionality from Angle and Norm Concentration") in [CCB+14]. Additionally, in [CCCR15], the same team of authors compared their approach with seven other selected estimation methods in a comprehensive benchmark framework. We choose "DANCo" for our numerical comparison since it outperforms all of its alternatives presented by Campadelli et al. in [CCC+11], [CCL+11], and [CCC+12]. From now on, to keep our upcoming presentations of results more compact, we substitute the abbreviation DANCo by "ANC" (Angle and Norm Concentration).

While the previous ID estimators (CD, GCD, MLE, DD) solely rely on information extracted from local distances, ANC utilizes both local distance and local angle infor-

mation. This is achieved by minimizing the sum of two Kullback-Leibler (in short: KL) divergences in order to obtain the final estimate.

Before we describe the assumptions of the approach, let us first give a short explanation of the term *Kullback-Leibler divergence*. Introduced in [KL51] as a measure of divergence or dissimilarity between two different probability distributions, it has become a quasi-standard for this purpose and is also called *information gain* or *relative entropy* in different contexts. Given two probability distributions P and Q on $\mathbb{R}^D$ with corresponding densities p and q, the KL divergence of Q from P is defined as

$$\mathcal{D}_{\mathrm{KL}}(P||Q) = \int_{\mathbb{R}^D} p(\boldsymbol{x}) \log \frac{p(\boldsymbol{x})}{q(\boldsymbol{x})} d\boldsymbol{x}. \tag{3.67}$$

It should be noted here, that some mild assumptions on p and q are required for the definition to be well-defined. Further, the KL divergence is not symmetric in terms of P and Q and can also be interpreted as the proportion of information that is lost when Q, typically a model, is used to approximate P, the "true" or measured distribution.

We now return to the setting of the ANC method. Given an m-dimensional manifold $M \subset \mathbb{R}^D$, embedded via some locally isometric smooth map, the data $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ are assumed to be samples of a probability measure P with corresponding smooth PDF $f : M \to \mathbb{R}_+$. In [CCB+14], the authors show that even if f is non-uniform, it is still locally uniform at $\boldsymbol{x}$, provided that $f(\boldsymbol{x}) \neq 0$. This result serves as justification to presume the density as constant in a small ball $B_r(\boldsymbol{x})$ of radius r around $\boldsymbol{x}$, as is also done in the preconditions of the MLE estimator.

Let us now introduce the first of the two Kullback-Leibler divergences, which is associated with the local distances. To this end, without loss of generality, consider the origin $\mathbf{0} \in \mathbb{R}^d$ and k points $\boldsymbol{z}_{i=1,\ldots,k}$ uniformly drawn from the corresponding unit ball $B_1(\mathbf{0}) \subset \mathbb{R}^d$. Next, consider the random variable defined by $r = \min_{i \in \{1,\ldots,k\}} \|\boldsymbol{z}_i\|$, i.e., the minimum distance r of all points to the origin. Then, it can be shown that the associated probability density function is given by

$$g_k(r; d) = k d r^{d-1} (1 - r^d)^{k-1}. \tag{3.68}$$

Note that when it comes to the final estimation procedure, d will be the intrinsic dimension to be estimated, while k, the number of nearest neighbors, will be the sole global parameter of the method. The above PDF describes the case where the smallest NN distance (of the k points to the origin) equals r, while the remaining $k-1$ NN distances are within the interval $(r, 1)$.

Given our dataset $\mathcal{X} = \{\boldsymbol{x}_{i=1,\ldots,N}\}$, consider now some point $\boldsymbol{x}_i$ and its $k+1$ nearest neighbors. In order to fit these points into the model, normalize all NN distances of $\boldsymbol{x}_i$ with $T_{k+1}(\boldsymbol{x}_i)$. The random variable r introduced above now corresponds to the normalized (minimum) distance $\rho(\boldsymbol{x}_i) := T_1(\boldsymbol{x}_i)/T_{k+1}(\boldsymbol{x}_i)$. Next, utilizing these distances $\rho(\boldsymbol{x}_i)$ in the density function of (3.68), the authors evaluate the log-likelihood with respect

to all N points as

$$\begin{aligned}\mathcal{L}_{\mathcal{X}}(d) &:= \sum_{i=1}^{N} \log g_k\left(\rho(\boldsymbol{x}_i); d\right) \\ &= N\log(kd) + (d-1)\sum_{i=1}^{N}\log\rho(\boldsymbol{x}_i) + (k-1)\sum_{i=1}^{N}\log\left(1-\rho^d(\boldsymbol{x}_i)\right).\end{aligned} \tag{3.69}$$

Subsequently, the optimization problem

$$\hat{m}_{\mathcal{X}} = \underset{d\in[1,D]}{\arg\max}\,\mathcal{L}_{\mathcal{X}}(d) \tag{3.70}$$

is solved numerically, where it is important to note that the search interval $[1, D]$ allows for a non-integer solution $\hat{m}_{\mathcal{X}}$. This result can now be plugged into (3.68) to yield the corresponding PDF $g_k(\,\cdot\,;\hat{m}_{\mathcal{X}})$ associated with the distances of the given data $\mathcal{X}$.

Eventually, the Kullback-Leibler divergence is used to measure the dissimilarity between the latter PDF and similar "reference" PDFs of known intrinsic dimensionality. Those reference PDFs are computed in the following way: for each for $d = 1, \ldots, D$, a set of N random points is sampled from the d-dimensional unit-ball, and the same maximum likelihood procedure as described above is applied to each point set; this yields associated dimension estimates $\check{m}_d$. Note that due to the bias of the estimator, in general, we have $\check{m}_d \neq d$ and thus, it is reasonable to work with the estimated values instead of the true underlying dimensions d. Finally, the authors of [CCB+14] derive an analytical expression for the Kullback-Leibler divergence

$$\mathrm{KL}_d^{(1)} = \mathcal{D}_{\mathrm{KL}}\left(g_k(\,\cdot\,;\hat{m}_{\mathcal{X}}) \,\middle\|\, g_k(\,\cdot\,;\check{m}_d)\right). \tag{3.71}$$

In a similar way, a second KL divergence — associated with local angles — is constructed. To this end, the authors employ the so-called *von Mises distribution* $\mathbb{M}(\mu,\kappa)$ (also known as *Tikhonov distribution*) with its probability density function

$$q(\theta;\mu,\kappa) = \frac{1}{2\pi I_0(\kappa)} e^{\kappa\cos(\theta-\mu)}, \tag{3.72}$$

where I_0 denotes the modified Bessel function of the first kind and order 0:

$$I_0(\kappa) = \frac{1}{2\pi}\int_0^{2\pi} e^{\kappa\cos\theta} d\theta. \tag{3.73}$$

Introduced by Richard von Mises in 1918, the aforementioned distribution can be viewed as the circular counterpart of the normal distribution on a line, where the two parameters in (3.72) are called *mean* μ and *concentration* κ. Further, μ and κ^{-1} are the respective analoga of the mean μ and variance σ^2 of the standard normal distribution $\mathcal{N}(\mu,\sigma^2)$. Now, for the final objective of the dimension estimation, the authors of the

ANC method seek to exploit the following relationship between the parameter κ and the dimension d. First, for angles $\theta \in [-\pi, \pi]$, the scaled random variable $\tilde{\theta} = \sqrt{d}(\theta - \pi/2)$ converges in distribution to $\mathcal{N}(0,1)$ for $d \to \infty$, as shown e.g. in [Söd11]. Second, as indicated above, for large values of κ, the von Mises distribution $\mathbb{M}(\mu, \kappa)$ can be approximated by $\mathcal{N}(\mu, \kappa^{-1})$. Combining these two results, it can be concluded that for $d \to \infty$, the concentration parameter κ converges asymptotically to the dimension d, meaning that $\lim_{d\to\infty}(\kappa/d) = 1$.

This motivates a similar procedure as before, using a Kullback-Leibler divergence. To this end, given an appropriate sample of pairwise angles $(\theta_1, \ldots, \theta_N)$, a maximum likelihood approach is used in order to yield parameter estimates $\hat{\mu}$ and $\hat{\kappa}$ for the mean and concentration. The details that lead to those estimates can be found in [CCB+14] and are not relevant for us.

Now, given a data point $\boldsymbol{x}_i$ and its associated k nearest neighbors $\boldsymbol{x}_i^{(1)}, \ldots, \boldsymbol{x}_i^{(k)}$, all possible $\binom{k}{2}$ angles between two of the k vectors $\left(\boldsymbol{x}_i - \boldsymbol{x}_i^{(j)}\right)$ are computed and accumulated in the set Θ_i. For each $\boldsymbol{x}_i$ and its set of angles Θ_i, the corresponding estimates $\hat{\mu}_i$ and $\hat{\kappa}_i$ are computed via the ML approach, and then, their respective averages $\hat{\mu}_{\mathcal{X}}$ and $\hat{\kappa}_{\mathcal{X}}$ are used in the KL divergence in an analogous manner as for the distances. For this purpose, again, random sample points from d-balls ($d = 1, \ldots, D$) are used to yield parameter estimates $\check{\mu}_d$ and $\check{\kappa}_d$, and the second Kullback-Leibler divergence is given by

$$\mathrm{KL}_d^{(2)} = \mathcal{D}_{\mathrm{KL}}\left(q(\,\cdot\,; \hat{\mu}_{\mathcal{X}}, \hat{\kappa}_{\mathcal{X}}) \;\middle\|\; q(\,\cdot\,; \check{\mu}_d, \check{\kappa}_d)\right). \tag{3.74}$$

In section 3.3 of [CCB+14], the authors discuss the challenges of combining the two KL divergences (3.71) and (3.74) into a single estimator. First, they remark that both approaches, if applied on their own, produce results with a considerable, but opposite bias. The main reason for this behavior should be the same as for many other dimension estimation methods, namely the use of a relatively small number of nearest neighbors k, whereas the model assumptions rely on a rather large number of input points. A possible solution to this intrinsic problem is not reconsidered by the authors of the ANC method. Instead, it is argued that since the distribution of distances $g_k(r; d)$ and the distribution of angles $q(\theta; \mu, \kappa)$ are independent for the setting of points sampled uniformly from d-balls, the joint probability density function factorizes and consequently, the KL divergence applied to the joint PDF corresponds exactly to the sum of both divergences. Therefore, the final dimension estimate is defined as

$$\hat{m} = \underset{d=1,\ldots,D}{\arg\min}\left[\mathrm{KL}_d^{(1)} + \mathrm{KL}_d^{(2)}\right], \tag{3.75}$$

where the exact analytic definitions of the divergences can be found in the original publication [CCB+14]. As mentioned above, the sole parameter of the estimator is the number k of nearest neighbors used in the computation of both KL divergences.

It is noteworthy that the authors propose an accelerated version of their method, named "FastDANCo". The acceleration is realized by precomputing the variables $\check{m}_d$, $\check{\mu}_d$

and $\check{\kappa}_d$, which do not depend on the precise data $\mathcal{X}$, but only on d and N, the number of input points. Once the parameter k is fixed, the three variables are computed for different values of d and N, in each case multiple times and averaged, and then the dependence of each variable on d and N is described using suitable fitting functions, or more specifically cubic smoothing splines. As reported by the authors, FastDANCo achieves results of similar quality as those of the original variant.

The ANC method is certainly an interesting approach since it is — at least to our knowledge — the only method combining local geometric information from distances and angles. Besides, it has been thoroughly tested and compared with other state-of-the-art approaches in both [CCB+14] and [CCCR15], where the authors claim the superiority of their technique.

Simplex Skewness (ESS)

A further approach introduced recently in [JSF15] by Johnsson, Soneson and Fontes is called "Expected Simplex Skewness" (ESS). Its model assumptions are not specified in detail. Instead, the authors presume that the data points are sampled from some sufficiently smooth m-dimensional manifold, where the sampling is dense enough and the influence of noise must not be too large. Eventually, just as the MLE or ANC method, it is required that each examined local neighborhood of the input data resembles a uniform distribution on an m-dimensional ball.

In a nutshell, the principal idea of the ESS method is the following. First, fix a "test dimension" parameter d, which must not be larger than the (unknown) intrinsic dimension m and can be safely chosen as $d = 1$ in the standard case. Next, in each available local neighborhood, construct multiple $(d+1)$-dimensional simplices using data points as vertices, compute their associated skewnesses, and compare the average of all skewness values with the (theoretical) expected simplex skewness for all possible dimensions to yield a corresponding estimate. As opposed to most other IDE methods, the ESS approach is only described and evaluated for small local patches (with $N = 50$ points) of some larger global datasets. Nevertheless, a global dimension estimate can easily be found by averaging the local results.

The notion of skewness is described informally by the authors in the following way. Given a set of local data points, they consider a simplex S defined by its vertices, the first of which is the centroid of all given points, the other $(d+1)$ are randomly sampled from the data. Then, the skewness of S is determined as its volume divided by "the volume it would have had if the edges incident to the centroid vertex were orthogonal" [JSF15]. The exact definition is only given for the case $d = 1$ as $\sin\theta$, where θ denotes the angle between the two edges incident to the centroid vertex. However, the *expected simplex skewness* is defined for uniformly distributed data in the unit n-ball $\mathcal{B}_n := B_1(\mathbf{0}) \subset \mathbb{R}^n$

for $d = 1$ and $d \geq 2$, respectively, as

$$s_n^{(1)} = \frac{1}{V_n} \int_{\mathcal{B}_n} |\sin \theta(\boldsymbol{x})| d\boldsymbol{x}, \tag{3.76}$$

$$s_n^{(d)} = \frac{1}{V_n^d} \int_{\mathcal{B}_n \times \cdots \times \mathcal{B}_n} \left| \boldsymbol{u} \wedge \frac{\boldsymbol{v}_1}{\|\boldsymbol{v}_1\|} \wedge \cdots \wedge \frac{\boldsymbol{v}_d}{\|\boldsymbol{v}_d\|} \right| d\boldsymbol{v}_1 \ldots d\boldsymbol{v}_d. \tag{3.77}$$

Here, $V_n = \pi^{n/2}/\Gamma(n/2+1)$ is the volume of $\mathcal{B}_n$. Next, $\theta(\boldsymbol{x})$ is defined as the angle between a fixed coordinate axis and the vector $\overrightarrow{\mathbf{0}\boldsymbol{x}}$. Finally, $\boldsymbol{u}$ denotes the unit vector along some reference coordinate axis, and the operator $\wedge$ represents the so-called *exterior product* or *wedge product*.

The exterior product is an algebraic construction used in Euclidean geometry to study n-dimensional volumes. The product of n vectors $\boldsymbol{v}_1 \wedge \boldsymbol{v}_2 \wedge \cdots \wedge \boldsymbol{v}_n$ is an object of a space also known as the *nth exterior power* and is sometimes called an *n-blade*. It can be interpreted as the oriented volume of the parallelotope spanned by the respective vectors.

In the supplemental material to [JSF15], the authors also derive the following closed expression for the expected simplex skewness (3.77):

$$s_n^{(d)} = \frac{\Gamma\left(\frac{n}{2}\right)^{d+1}}{\Gamma\left(\frac{n+1}{2}\right)^d \Gamma\left(\frac{n-d}{2}\right)}. \tag{3.78}$$

Consequently, for fixed d, we have monotone convergence $s_n^{(d)} \nearrow 1$ for $n \to \infty$. The first few values of $s_n^{(1)}$ and $s_n^{(2)}$ can be found in table 3.2.

n	2	3	4	5	6	7	8	9	10	11	12
$s_n^{(1)}$	0.637	0.785	0.849	0.884	0.905	0.920	0.931	0.940	0.946	0.951	0.956
$s_n^{(2)}$	–	0.393	0.566	0.663	0.724	0.767	0.798	0.822	0.841	0.856	0.869

Table 3.2: Numerical values (rounded to 3 digits) of the expected simplex skewness for $d = 1, 2$ and $n = 2, \ldots, 12$.

As noted above, the authors only consider local patches of the data, which are supposed to resemble a uniform distribution on some m-dimensional ball. Given such a dataset $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$, the entire estimation procedure, for the parameter d fixed as $d = 1$, can be summarized as follows. First, the data are centered such that their centroid coincides with the origin. If the total number of different simplices does not exceed $C = 5000$, the skewness of each simplex determined by the origin and $(d+1)$ data points is computed, otherwise, C simplices are chosen at random amongst all available possibilities. Next, a weighted average of all skewness values is computed. Since the

relative impact of noise is in general larger for groups of points forming smaller simplices, the authors choose the weight for each simplex equal to the product of the lengths of the edges incident to the centroid. Thus, the estimator of the simplex skewness (for $d = 1$) is given by

$$\hat{s}^{(1)} = \frac{\sum_{i \neq j} |\overline{\boldsymbol{x}}_i \wedge \overline{\boldsymbol{x}}_j|}{\sum_{i \neq j} \|\overline{\boldsymbol{x}}_i\| \cdot \|\overline{\boldsymbol{x}}_j\|}, \tag{3.79}$$

where $\overline{\boldsymbol{x}}_i$ denotes the vector from $\mathbf{0}$ to the centered data point $\tilde{\boldsymbol{x}}_i$. The final dimension estimate is a real number determined via interpolation. For this purpose, and for general $d \geq 1$, if n is chosen such that $s_n^{(d)} \leq \hat{s}^{(d)} < s_{n+1}^{(d)}$, the dimension estimate is given by

$$\hat{m} = n + \frac{\hat{s}^{(d)} - s_n^{(d)}}{s_{n+1}^{(d)} - s_n^{(d)}}. \tag{3.80}$$

The ESS method as described in [JSF15] features only a single parameter d which is required to satisfy $d < m$, but can be set to 1 as default option. In their experiments, the authors compare the algorithm for $d = 1$ and $d = 2$ and conclude that there is no significant superiority of one or the other variant. Since the choice of $d = 1$ is both easier to implement and more flexible (it is able to distinguish between datasets of intrinsic dimensionality of $m = 1$ and $m = 2$), we thus restrict our upcoming numerical considerations to this variant.

In practice, the ESS approach naturally requires a particular procedure to choose the local datasets. In the simplest case, a second parameter k is introduced, and for each point $\boldsymbol{x}_i$ of the global dataset $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$, the local dataset consisting of the k nearest neighbors of $\boldsymbol{x}_i$ is taken as input for the original ESS method, and, in the last step, the local results are averaged.

For the sake of completeness, we mention that the authors of [JSF15] propose a second version referred to as *ESSb* as opposed to *ESSa*, which denotes the method described here. In the *ESSb* approach, given two normalized vectors constructed from the data points as explained above, the estimated quantity is the expected length of the projection of the first onto the second vector. The expectation value of the new quantity for $d = 1$ is then given by formula (3.76), where the term $\sin\theta$ is replaced by $\cos\theta$. Since the corresponding technique yields similar results, we only consider the original ESS method in the following.

As mentioned above, the ESS method is interesting for our purposes, as it is related to our own approach and it has not been examined in another publication known to us. In the basic case where $d = 1$ (let us abbreviate this by $\mathrm{ESS}^{(1)}$), only angles between vectors given by data points are analyzed, hence, it relies on concentration effects of angles in high dimensions, similar to the ANC method. According to the authors, the ESS technique is also able to reliably estimate intrinsic dimension values m exceeding the number of points N for certain datasets.

3.3 Simplex Volume Computation

In contrast to all methods for intrinsic dimension estimation presented above, our approach relies on the evaluation of numerous potentially high-dimensional simplex volumes, where the corresponding vertex points have been drawn from some local point subset $\mathcal{S} \subset \mathcal{X} \subset \mathbb{R}^D$ of moderate size in each case. More precisely, the dimension of the simplices can be as high as the intrinsic dimension m of the dataset $\mathcal{X}$. While the computation of Euclidean distances and angles is generally a trivial task, the efficient determination of n-dimensional simplex volumes for $n \gg 3$ is significantly more challenging.

In the context of dimension estimation and dimension reduction, one naturally assumes that the intrinsic dimension is considerably smaller than the ambient dimension, i.e., $m \ll D$. In this section, we therefore present a technique for the efficient volume evaluation of multiple n-dimensional simplices with vertex points in $\mathbb{R}^D$, where, after a pre-computation step of $\mathcal{O}(D)$ time complexity, each simplex volume computation can be performed with only $\mathcal{O}(n^3)$ operations, independently of the ambient dimension D in a numerically stable way. In order to explain this procedure we require a few basic results of Euclidean geometry.

3.3.1 Theoretical preliminaries

Our introduction to some fundamental results of Euclidean geometry follows in parts the works of Berger, see [Ber09a], [Ber09b]. As commonplace, let $\langle \cdot, \cdot \rangle : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$ denote the standard inner product of the Euclidean space $\mathbb{R}^D$, with the associated norm $\| \cdot \| : \mathbb{R}^D \to \mathbb{R}$. Further, for $\boldsymbol{z}_i, \boldsymbol{z}_j \in \mathbb{R}^D$, the Euclidean distance of the two elements is sometimes abbreviated as $\delta_{ij} := \|\boldsymbol{z}_i - \boldsymbol{z}_j\|$.

Affine subspaces

An *affine subspace* S of the vector space $\mathbb{R}^D$ is a subset closed under affine combinations of its elements. Formally, for arbitrary elements $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n \in S$ and scalars $\alpha_1, \ldots, \alpha_n \in \mathbb{R}$ with $\sum_{i=1}^n \alpha_i = 1$, the linear combination $\sum_{i=1}^n \alpha_i \cdot \boldsymbol{z}_i$ is also contained in S. The affine subspace S is equivalent to a linear subspace translated away from the origin and thus, can be written as $S = \boldsymbol{p} + W$, where $\boldsymbol{p}$ can be any element of S, and W is the corresponding linear subspace. The dimension of S is inherited from W. Zero-, one-, and two-dimensional affine subspaces are called *points*, *lines*, and *planes*, respectively. In the following, if unambiguous, we will refer to "affine subspace" simply as "subspace".

Parallelotopes and simplices

Let $2 \leq n \leq D$. A set of $n+1$ points $\boldsymbol{z}_0, \ldots, \boldsymbol{z}_n \in \mathbb{R}^D$ is called *affinely independent*, if there is no $(n-1)$-dimensional affine subspace $S \subset \mathbb{R}^D$ with $\boldsymbol{z}_0, \ldots, \boldsymbol{z}_n \in S$. The

parallelotope $\Pi = \Pi(\boldsymbol{z}_0, \ldots, \boldsymbol{z}_n)$ determined by a set of affinely independent points is defined as

$$\Pi = \left\{ \boldsymbol{z} \in \mathbb{R}^D \;\middle|\; \boldsymbol{z} = \boldsymbol{z}_0 + \sum_{i=1}^{n} \alpha_i \cdot \overrightarrow{\boldsymbol{z}_0 \boldsymbol{z}_i}, \;\; \text{where} \;\; 0 \leq \alpha_i \leq 1 \right\}, \tag{3.81}$$

where $\overrightarrow{\boldsymbol{z}_0 \boldsymbol{z}_i} := \boldsymbol{z}_i - \boldsymbol{z}_0$. This parallelotope is also called n-parallelotope, since it is contained in an n-dimensional subspace. A 2-parallelotope is a parallelogram, a 3-parallelotope is known as parallelepiped. The *simplex* $\Delta = \Delta(\boldsymbol{z}_0, \ldots, \boldsymbol{z}_n)$ determined by the same point set is given by its associated convex hull:

$$\Delta = \left\{ \boldsymbol{z} \in \mathbb{R}^D \;\middle|\; \boldsymbol{z} = \sum_{i=0}^{n} \alpha_i \cdot \boldsymbol{z}_i, \;\; \text{where} \;\; 0 \leq \alpha_i \leq 1 \;\; \text{and} \;\; \sum_{i=0}^{n} \alpha_i = 1 \right\}. \tag{3.82}$$

Correspondingly, we call it an n-simplex. A 2-simplex is a triangle, a 3-simplex is a tetrahedron. We can extend our definition to a 1-simplex as a line between two points, and a 0-simplex as a single point. Since every nonempty subset of the $n+1$ points $\boldsymbol{z}_0, \ldots, \boldsymbol{z}_n$ is also affinely independent, its convex hull is again a simplex, which we call *face* of the simplex Δ. The points $\boldsymbol{z}_i$, the faces of dimension 0, are called *vertices*. The one-dimensional faces are called *edges*, and the $(n-1)$-dimensional faces are called *facets* of the simplex. The same terms (i.e. vertex, edge, facet) can be defined in an analogous manner for the parallelotope.

Volume in Euclidean space

The canonical measure for volume in the Euclidean space $\mathbb{R}^D$ is the Lebesgue measure $\mu : \mathcal{L}(\mathbb{R}^D) \to \overline{\mathbb{R}}$, where $\mathcal{L}(\mathbb{R}^D)$ is the set of Lebesgue-measurable sets, and $\overline{\mathbb{R}}$ is the extended real number system $\mathbb{R} \cup \{-\infty, +\infty\}$. For a comprehensive introduction into measure theory, we refer to [Rud87]. The volume of a compact set $K \subset \mathbb{R}^D$ is now defined as

$$\mathrm{vol}(K) = \int_{\mathbb{R}^D} \chi_K \mu, \tag{3.83}$$

where χ_K is the characteristic function of K.

Let us again consider a set of $n+1$ affinely independent points $\boldsymbol{z}_0, \ldots, \boldsymbol{z}_n \in \mathbb{R}^D$ and let $\boldsymbol{v}_i = \overrightarrow{\boldsymbol{z}_0 \boldsymbol{z}_i} = \boldsymbol{z}_i - \boldsymbol{z}_0, (i = 1, \ldots, n)$. In [Ber09a], it is shown that the volume of the parallelotope $\Pi = \Pi(\boldsymbol{z}_0, \ldots, \boldsymbol{z}_n)$ can be calculated via the determinant of the Gram matrix as

$$\mathrm{vol}(\Pi) = \sqrt{\det\left(\mathrm{Gram}\left(\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n\right)\right)}, \tag{3.84}$$

where the *Gram matrix* is given by

$$\operatorname{Gram}(\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n) = \begin{pmatrix} \langle \boldsymbol{v}_1, \boldsymbol{v}_1 \rangle & \ldots & \langle \boldsymbol{v}_1, \boldsymbol{v}_n \rangle \\ \vdots & & \vdots \\ \langle \boldsymbol{v}_n, \boldsymbol{v}_1 \rangle & \ldots & \langle \boldsymbol{v}_n, \boldsymbol{v}_n \rangle \end{pmatrix}. \tag{3.85}$$

Furthermore, the volume of the corresponding simplex $\Delta = \Delta(\boldsymbol{z}_0, \ldots, \boldsymbol{z}_n)$ is given by

$$\operatorname{vol}(\Delta) = \frac{1}{n!} \operatorname{vol}(\Pi) = \frac{1}{n!} \sqrt{\det(\operatorname{Gram}(\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n))}. \tag{3.86}$$

Hence, the Gram matrix is the key to volume computation of basic geometric objects. Consequently, let us recall some of its important properties. First, the Gram matrix is always positive semidefinite, and it is positive definite if and only if its defining vectors $\boldsymbol{v}_i$ are linearly independent. The Gram determinant can be bounded via the following inequality (cf. [Fis00]).

Theorem 3.1 (Hadamard's inequality)**.** *For $n < D$, let $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n \in \mathbb{R}^D$ be arbitrary vectors. Then the following inequality holds:*

$$0 \leq \det(\operatorname{Gram}(\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n)) \leq \prod_{i=1}^{n} \|\boldsymbol{v}_i\|^2, \tag{3.87}$$

where $\det(\operatorname{Gram}(\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n)) = 0$ *if and only if the vectors $\boldsymbol{v}_i$ are linearly dependent.*

The geometric intuition behind this bound is straightforward: the parallelotope features its maximal volume in case all $\boldsymbol{v}_i$ are pairwise orthogonal and thus form a box, while the volume of the box simply equals the product of the vector lengths.

Distance relationships of affinely dependent points

Given a set of points $\boldsymbol{z}_0, \ldots, \boldsymbol{z}_n \in \mathbb{R}^D$ $(n \geq 2)$ in the same $(n-1)$-dimensional affine subspace S, it is reasonable to assume that there is some relationship between their pairwise distances $\delta_{ij} := \|\boldsymbol{z}_i - \boldsymbol{z}_j\|$. Consider for instance three points $\boldsymbol{z}_0, \boldsymbol{z}_1, \boldsymbol{z}_2$ on a line. The corresponding distances satisfy the equation

$$(\delta_{01} + \delta_{02} - \delta_{12}) \cdot (\delta_{01} + \delta_{12} - \delta_{02}) \cdot (\delta_{02} + \delta_{12} - \delta_{01}) = 0, \tag{3.88}$$

since the sum of the two smaller distances equals the largest distance. Indeed, a similar relation exists for the general case $n \geq 2$. The underlying idea is that the volume of the appropriate (degenerated) parallelotope (or simplex, respectively) must be zero and can be calculated using nothing but the pairwise distances. For this purpose, we introduce the

Definition 3.2 (Cayley-Menger determinant)**.** *Given $n+1$ points $\boldsymbol{z}_0, \ldots, \boldsymbol{z}_n \in \mathbb{R}^D$ with pairwise distances $\delta_{ij} = \|\boldsymbol{z}_i - \boldsymbol{z}_j\|$, we define the* Cayley-Menger determinant *as*

$$\Lambda(\boldsymbol{z}_0, \ldots, \boldsymbol{z}_n) = \begin{vmatrix} 0 & 1 & 1 & \ldots & 1 \\ 1 & 0 & \delta_{01}^2 & \ldots & \delta_{0n}^2 \\ 1 & \delta_{10}^2 & 0 & & \delta_{1n}^2 \\ \vdots & \vdots & & \ddots & \vdots \\ 1 & \delta_{n0}^2 & \delta_{n1}^2 & \ldots & 0 \end{vmatrix}. \tag{3.89}$$

The following relationship is now shown in [Ber09a]:

$$\det\left(\mathrm{Gram}\left(\overrightarrow{\boldsymbol{z}_0\boldsymbol{z}_1}, \ldots, \overrightarrow{\boldsymbol{z}_0\boldsymbol{z}_n}\right)\right) = \frac{(-1)^{n+1}}{2^n} \cdot \Lambda(\boldsymbol{z}_0, \ldots, \boldsymbol{z}_n). \tag{3.90}$$

Thus, we have a convenient criterion to decide whether a set of $n+1$ points with given pairwise distances are contained in the same affine subspace.

Theorem 3.3. *Let $\boldsymbol{z}_0, \ldots, \boldsymbol{z}_n \in \mathbb{R}^D$ be arbitrary points with pairwise Euclidean distances $\delta_{ij} = \|\boldsymbol{z}_i - \boldsymbol{z}_j\|$. Then a necessary and sufficient condition for the points to be affinely dependent (i.e. contained in an $(n-1)$-dimensional affine subspace) is*

$$\Lambda(\boldsymbol{z}_0, \ldots, \boldsymbol{z}_n) = 0. \tag{3.91}$$

Furthermore, combining relationship (3.90) with (3.86) leads us to another formula for the volume of the simplex $\Delta = \Delta(\boldsymbol{z}_0, \ldots, \boldsymbol{z}_n)$:

$$\mathrm{vol}(\Delta) = \frac{1}{n!}\sqrt{\frac{(-1)^{n+1}}{2^n} \cdot \Lambda(\boldsymbol{z}_0, \ldots, \boldsymbol{z}_n)}. \tag{3.92}$$

This is an interesting geometrical result, since the simplex volume can be calculated without knowledge of the exact vertex coordinates — only the distances between the vertices are required. From the numerical viewpoint, the computation of the Cayley-Menger determinant appears to be troublesome, since the diagonal consists of nothing but zeros. Therefore, we derive another formula for the simplex volume.

For given $\boldsymbol{z}_0, \ldots, \boldsymbol{z}_n \in \mathbb{R}^D$ and their distances δ_{ij}, we again consider the vectors $\boldsymbol{v}_i = (\boldsymbol{z}_i - \boldsymbol{z}_0)$ or rather the normalized vectors $\overline{\boldsymbol{v}}_i = (\boldsymbol{z}_i - \boldsymbol{z}_0)/\delta_{0i}$ for $i = 1, \ldots, n$. Now the Gram matrix associated with the $\overline{\boldsymbol{v}}_i$ turns out to have a nice structure, which motivates another definition.

Definition 3.4 (Spherical Cayley-Menger determinant)**.** *Let $\boldsymbol{z}_i \in \mathbb{R}^D$ with distances δ_{ij} and $\overline{\boldsymbol{v}}_i \in \mathbb{R}^D$ be as above. Then we define the* spherical Cayley-Menger determinant *as*

the Gram determinant

$$\widetilde{\Lambda}\left(\boldsymbol{z}_0,\ldots,\boldsymbol{z}_n\right)=\det\left(\mathrm{Gram}\left(\overline{\boldsymbol{v}}_1,\ldots,\overline{\boldsymbol{v}}_n\right)\right)=\begin{vmatrix}1 & c_{12} & c_{13} & \ldots & c_{1n}\\ c_{21} & 1 & c_{23} & \ldots & c_{2n}\\ c_{31} & c_{32} & 1 & & \vdots\\ \vdots & \vdots & & \ddots & c_{n\text{-}1,n}\\ c_{n1} & c_{n2} & \ldots & c_{n,n\text{-}1} & 1\end{vmatrix}. \tag{3.93}$$

Here, $c_{ij}:=\langle\overline{\boldsymbol{v}}_i,\overline{\boldsymbol{v}}_j\rangle=\cos(\measuredangle(\overline{\boldsymbol{v}}_i,\overline{\boldsymbol{v}}_j))$ *can also be written in terms of distances as*

$$c_{ij}=\frac{\delta_{0i}^2+\delta_{0j}^2-\delta_{ij}^2}{2\delta_{0i}\delta_{0j}}.$$

Finally, since the determinant is linear in each row and each column, we have

$$\det\left(\mathrm{Gram}\left(\boldsymbol{v}_1,\ldots,\boldsymbol{v}_n\right)\right)=\prod_{i=1}^{n}\delta_{0i}^2\cdot\det\left(\mathrm{Gram}\left(\overline{\boldsymbol{v}}_1,\ldots,\overline{\boldsymbol{v}}_n\right)\right),$$

and thus, we can rewrite (3.86) as

$$\mathrm{vol}(\Delta)=\frac{\prod_{i=1}^{n}\delta_{0i}}{n!}\sqrt{\widetilde{\Lambda}\left(\boldsymbol{z}_0,\ldots,\boldsymbol{z}_n\right)}. \tag{3.94}$$

Thus, we derived a second formula for the simplex volume that uses nothing but distances between the vertices. As mentioned before, the Gram matrix of linearly independent vectors is positive definite, which will simplify the numerical evaluation of the Gram determinant. Therefore, for our purposes, formula (3.94) seems to be more favorable as opposed to (3.92).

For the reader interested in Cayley-Menger type determinants and their geometric interpretations, further examples can be found in [Ber09a], [Aud11] and [MF04].

3.3.2 Efficient numerical computation of simplex volumes

We are now ready to describe the procedure for the rapid evaluation of multiple simplex volumes. For a given local point set of moderate size, in the first step, all inter-point distances are pre-computed and stored. In the second step, the volume of a particular simplex with vertices selected from this point set is calculated via equation (3.94) using nothing but distances; here, the exact coordinates of the D-dimensional vertex points are not required any more. It still remains to provide an efficient way to evaluate the spherical Cayley-Menger determinant $\widetilde{\Lambda}\left(\boldsymbol{z}_0,\ldots,\boldsymbol{z}_n\right)$, see eq. (3.93). To this end, we proceed with some condensed remarks on determinants.

The concept of the determinant of a matrix $A\in\mathbb{R}^{n\times n}$ can be motivated in various ways. The geometric interpretation as the oriented volume of a parallelotope has been

used in the above considerations. The classical algebraic version is the so-called *Leibniz formula*

$$\det(A) = \sum_{\sigma \in S_n} \operatorname{sgn}(\sigma) \prod_{i=1}^{n} a_{\sigma(i),i}, \tag{3.95}$$

defining the determinant via permutations σ of the matrix elements $a_{i,j}$. Finally, WEIERSTRASS gave an elegant axiomatic description of the determinant function using only three axioms, see e.g. [Fis00].

The numerical computation of a matrix determinant is in fact closely connected with the solution of a linear equation system. Clearly, the direct evaluation of the Leibniz formula requires $\mathcal{O}(n!)$ operations and thus gets infeasible in practice even for moderate values of n. In contrast, a classical LU decomposition via Gaussian elimination has a complexity of only $\mathcal{O}(n^3)$ and provides a straightforward way to compute the determinant as

$$\det(A) = \det(LU) = \det(L) \cdot \det(U) = \det(U) = \prod_{i=1}^{n} u_{ii},$$

where $\det(L) = 1$, since its diagonal elements are all ones. More precisely, LU decomposition with partial pivoting requires a permutation matrix P to reorder the rows of A and yields a factorization $PA = LU$ for any square matrix A. The determinant for this general case is then given by

$$\det(A) = (-1)^r \det(U), \tag{3.96}$$

where r is the number of row permutations in P. The same procedure can be adopted for the QR decomposition, since $\det(Q) = \pm 1$ for any orthogonal matrix Q. An error analysis and numerical results for implementations of these decomposition approaches can be found in [PYS97]. Beyond that, many efforts have been made to treat special cases of determinants in an optimized way. Division-free algorithms for integer matrices have been investigated e.g. in [Sam42], [Bar68], [Kal92] and [Rot01]. For very large matrices, an approach presented in [KV05] based on the CoppersmithWinograd algorithm for fast matrix multiplication (see [CW90]) reduces the complexity from $\mathcal{O}(n^3)$ to $\mathcal{O}(n^{2.7})$. Certainly, the determinants of sparse or block matrices can be evaluated with suitable efficient algorithms.

Our method requires the evaluation of Gram determinants of normed vectors as presented in definition 3.93. As mentioned above, the corresponding Gram matrix is always positive semidefinite. Thus, we can distinguish two cases: either the vectors defining the matrix are linearly independent and consequently, the matrix is positive definite with a positive determinant; or the vectors are linearly dependent resulting in a singular matrix with a determinant equal to zero.

Some fundamental properties of symmetric and positive definite matrices are summarized in the following theorem (see e.g. [HJ12]).

Theorem 3.5. *Let $A \in \mathbb{R}^{n \times n}$ be symmetric. Then each of the following conditions is*

equivalent for A being positive definite, that is $\boldsymbol{x}^T A\boldsymbol{x} > 0$ for all $\boldsymbol{x} \in \mathbb{R}^n, \boldsymbol{x} \neq 0$:

(i) Every principal submatrix $A^{(k)}$ (i.e. the upper-left $(k \times k)$-submatrix of A) is positive definite.

(ii) Every leading principal minor (i.e. the determinant of $A^{(k)}$) is positive.

(iii) Every eigenvalue of A is positive.

(iv) A has a unique Cholesky decomposition $A = LL^T$.

In particular, the Cholesky decomposition of a symmetric positive definite matrix requires no pivoting and is numerically stable according to [Sto05]. It involves roughly $n^3/3$ floating point operations and can be performed in-place, see e.g. [GVL96].

Summing up, the Cholesky decomposition is clearly to be favored — not only due to its practical advantages — as compared to specialized approaches mentioned above. The latter algorithms can usually only benefit from their slightly superior runtime performance for large matrices, but not for the matrices of moderate sizes we are dealing with.

Our proceeding for the rapid and efficient computation of multiple n-dimensional simplex volumes with vertex points drawn from some moderately sized point set $\mathcal{S} \subset \mathbb{R}^D$ can now be subsumed as follows.

(0) Pre-compute and store all inter-point distances $\delta_{ij} = \|\boldsymbol{z}_i - \boldsymbol{z}_j\|$ for $\boldsymbol{z}_i, \boldsymbol{z}_j \in \mathcal{S}$.

(1) For a given simplex $\Delta(\boldsymbol{z}_0, \ldots, \boldsymbol{z}_n)$: build the Gram matrix as in (3.93) using the corresponding distances δ_{ij}.

(2) Evaluate the determinant of the Gram matrix via Cholesky decomposition and apply formula (3.94) to yield the simplex volume.

The runtime complexity for a set of size $|\mathcal{S}| = k$ is $\mathcal{O}(D \cdot k(k-1)/2)$ for the pre-computation and $\mathcal{O}(n^3/3)$ for each simplex volume evaluation. The crucial fact is that the costs of each volume computation is independent of the ambient dimension D.

3.4 Intrinsic Dimension Estimation via Sample Simplex Volumes (SSV)

We now reached the point where we present our approach for intrinsic dimension estimation based on volumes of sample simplices, which is why we named it *Sample Simplex Volume* (SSV) method; to be exact, we will introduce two slightly different variants called SSV1 and SSV2 method.

In the first subsection, we provide a short motivation for the use of simplex volumes. In subsection 3.4.2, we present our underlying model assumptions and three theorems

from geometric probability that represent the foundation of our IDE concept. The main descriptions of our two approaches are given in subsection 3.4.3; both the SSV1 and the SSV2 method are motivated and the precise procedures are unveiled. Subsequently, a compact examination of the influence of noise on simplex volumes is given in subsection 3.4.4, right before the algorithmic description of both SSV methods in 3.4.5. Finally, we conclude this section with the complexity analysis in 3.4.6.

3.4.1 Why simplex volumes?

First, we would like to state the consideration behind our decision to rely on high-dimensional volumes contrary to most other approaches that are only based on pairwise distances (MLE, GCD, DD, and many more) or angular information (ANC and $\text{ESS}^{(1)}$). Consider a set of points $\mathcal{X} = \{\boldsymbol{x}_{i=1,\dots,N}\} \subset \mathbb{R}^D$ with $N \gg D$. A Euclidean distance $\|\boldsymbol{x}_i - \boldsymbol{x}_j\|$ reflects a relationship between two points, an angle (or alternatively, the area of a triangle) reflects a relationship between three points. Similarly, for $d \geq 2$, the volume of a d-simplex spanned by its $(d+1)$ vertices reflects a relationship between $(d+1)$ points.

Furthermore, using statistical techniques to analyze all pairwise distances between some $(d+1)$ points, there is no straightforward way to decide whether the points span an actual d-dimensional space or whether they belong to a lower dimensional subspace. However, as shown in subsection 3.3.1, the volume of the corresponding simplex is the answer to this question and also, in case it is non-zero, can provide a measure of the deviation from a potential lower dimensional subspace. Clearly, this information can also be derived from the pairwise distances, since the computation of the simplex volume requires nothing but those, as shown above. Summing up, our intuition is that the concept of analyzing high-dimensional volumes — being a more complex quantity than pairwise distances — might enable new and more advanced approaches of estimating the intrinsic dimensionality of datasets.

3.4.2 Model and theoretical background

We start with the description of our model. To begin with, our theoretical and algorithmic considerations are based on noisefree data. Subsection 3.4.4 will be dedicated to the subject of noise.

We assume that the data $\mathcal{X} = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_N\} \subset \mathbb{R}^D$ is sampled from some smooth m-dimensional manifold M, where $m < D$, and both the intrinsic dimension m and the precise parametric description of M are unknown. Consequently, we rely on the fact that our manifold locally resembles an m-dimensional affine space. In particular, we ignore the local curvature of M, i.e., we treat the linear approximation as perfectly linear. Therefore, we can suppose that, for any data point $\boldsymbol{x}_i$, all data points within a small ball $B_r(\boldsymbol{x}_i)$ (for some small, but unknown radius r) are part of or close to some m-dimensional affine space. According to an argumentation presented in [CCB+14]

(compare also the description of the ANC method in subsection 3.2.2), it is furthermore reasonable to assume that points within $B_r(\boldsymbol{x}_i)$ are approximately distributed according to the m-dimensional uniform distribution, even if the underlying global distribution function might be non-uniform.

Let us now make a short excursion into topics from geometric probability to provide the theoretical foundation for our method. More precisely, we present three different results on the expected volume of random simplices. The first result taken from [Kin69] is due to KINGMAN and can be summarized in the following

Theorem 3.6. *For some fixed $n \geq 1$, let $B_r^n(\mathbf{0})$ be a ball in $\mathbb{R}^n$ around the origin with radius $r > 0$ with n-dimensional volume $V_n(r)$. Let further $\boldsymbol{z}_0, \ldots, \boldsymbol{z}_n \in \mathbb{R}^n$ be random points drawn independently from a uniform distribution over the ball $B_r^n(\mathbf{0})$. The random n-simplex spanned by the vertex points $\boldsymbol{z}_0, \ldots, \boldsymbol{z}_n$ shall be denoted by Δ_n. Then, the expected value of the random volume of Δ_n is given by*

$$\mathbb{E}\left[\mathrm{vol}_n(\Delta_n)\right] = V_n(r) \cdot \binom{n+1}{\frac{1}{2}(n+1)}^{n+1} \cdot \left[\binom{(n+1)^2}{\frac{1}{2}(n+1)^2} \cdot 2^n\right]^{-1}. \tag{3.97}$$

For future reference, we define the following constants:

$$\nu(n) := \binom{n+1}{\frac{1}{2}(n+1)}^{n+1} \cdot \left[\binom{(n+1)^2}{\frac{1}{2}(n+1)^2} \cdot 2^n\right]^{-1}. \tag{3.98}$$

These describe the expected simplex volume for the case of a surrounding n-dimensional ball of volume 1. Some numerical values of $\nu(n)$ for small n can be found in table 3.3.

n	1	2	3	4	5	6	7	20	30
$\nu(n)$	0.333	0.0739	0.0126	$1.79e^{-3}$	$2.20e^{-4}$	$2.44e^{-5}$	$2.46e^{-6}$	$2.23e^{-21}$	$1.97e^{-34}$

Table 3.3: Numerical values (rounded to 3 digits) of the expected simplex volumes $\nu(n)$ with vertex points randomly drawn from within an n-dimensional ball with volume 1.

The general question of the expected volume of a simplex with vertices chosen randomly within a given convex body has been investigated in many different variants (see e.g. [Kle69] and [Mil71]) and might have its origin in the well-known *four point problem* by SYLVESTER [Pfi89], which is strongly related to the two-dimensional version of the problem considered here. The solution requires the (non-trivial) evaluation of multiple nested integrals. While many two-dimensional and some three-dimensional variants could be evaluated explicitly, a universal formula as given in theorem 3.6 has not been found — at least to our knowledge — for the important case where the surrounding convex body is an n-dimensional cube, compare also [CFG91]. Recent developments in

the research field of random polytopes and their volumes can be found e.g. in [Sch08] and [Hug13].

The second result presented by Miles (theorem 2 in [Mil71]) is basically a generalization of theorem 3.6. It provides a formula for the kth order moments of the random p-volume of the p-simplex whose $p+1 = s+t$ vertices consist of s points sampled from within the unit n-ball and t points sampled from the boundary, i.e., the unit $(n-1)$-sphere. For the sake of clarity, we recite the formula for $k = 1$ and $t = 0$ only, which is sufficient for our purposes. Instead of the unit ball, we consider the more general case of a given n-ball with radius $r > 0$.

Theorem 3.7. *For some fixed $n \geq 1$, let $B_r^n(\mathbf{0})$ be a ball in $\mathbb{R}^n$ around the origin with radius $r > 0$. For $1 \leq s \leq n$, let $s+1$ points be randomly and independently drawn from a uniform distribution over the ball $B_r^n(\mathbf{0})$. Then, the expected value of the random s-volume of the random s-simplex Δ_s spanned by these $s+1$ points is given by*

$$\begin{aligned}\mathbb{E}_n\left[\mathrm{vol}_s(\Delta_s)\right] &= \frac{r^s}{s!}\left(\frac{n}{n+1}\right)^{s+1}\frac{\Gamma\left(\frac{1}{2}(s+1)(n+1)+1\right)}{\Gamma\left(\frac{1}{2}((s+1)(n+1)+1)\right)} \\ &\quad\cdot\frac{\Gamma\left(\frac{1}{2}(n+1)\right)}{\Gamma\left(\frac{1}{2}(n-s+1)\right)}\left(\frac{\Gamma\left(\frac{1}{2}n\right)}{\Gamma\left(\frac{1}{2}(n+1)\right)}\right)^{s+1}.\end{aligned} \tag{3.99}$$

Naturally, letting $s = n$, the above formula coincides with equation (3.97), which is just a more compact representation. Again, for future reference, we define the following constants:

$$\xi(r,n,s) := \mathbb{E}_n\left[\mathrm{vol}_s(\Delta_s)\right] \tag{3.100}$$

as given by eq. (3.99).

The third result is due to Groemer [Gro73] and reveals the role of the n-ball as opposed to other convex bodies.

Theorem 3.8. *Let $K \subset \mathbb{R}^n$ be any convex body for some fixed $n \geq 1$. Let further $\Delta_{[K]}$ denote the random n-simplex with $n+1$ vertices randomly and independently drawn from a uniform distribution over K, and $\mathbb{E}[\mathrm{vol}_n(\Delta_{[K]})]$ denote the expected volume of $\Delta_{[K]}$. Then we have*

$$\mathbb{E}[\mathrm{vol}_n(\Delta_{[B]})] \leq \mathbb{E}[\mathrm{vol}_n(\Delta_{[K]})], \tag{3.101}$$

where B is the n-dimensional ball with the same volume as K. Equality holds if and only if K is an ellipsoid.

Let us offer the following intuition for this theorem. First, the problem of finding the convex body with the above minimal property is obviously invariant under volume preserving affine transformations, which is why we can consider the n-dimensional ball instead of arbitrary ellipsoids. Now note that, among all n-dimensional convex bodies, the n-simplex is the most "spiky" while the n-ball is the least spiky one. Thus, when

inscribing a simplex of maximal volume into a given ball, a lot of volume of the ball is wasted between the facets of the simplex and the surface of the ball. On the other hand, if we replace the ball by a simplex of the same volume, its interior can be perfectly filled by an inscribed simplex. This insight suggests that random simplices sampled from the interior of some n-ball should on average be smaller than when sampled from the interior of another convex object of the same volume.

While theorem 3.8 is basically cited for the sake of completeness and for deeper understanding, the remaining two theorems provide the justification and motivation for our approach which we present now.

3.4.3 Concept of the SSV approach

The main idea behind the SSV approach is simple: compute multiple volumes of random d-simplices with vertices sampled from local datasets and compare the average of these volumes with the expected values as given in theorem 3.6 and theorem 3.7, respectively. The basic proceeding of our IDE method and the rationale for its practicability, which might not be as obvious, will be presented below. In fact, we will introduce two slightly different versions of the SSV method, named SSV1 and SSV2. While the SSV1 method has first been developed for the ideal case of noisefree data, the SSV2 method is more suitable to handle moderate levels of noise and also has much lower runtime costs when it comes to data with higher intrinsic dimension m. In the following, we introduce and motivate the approach of both SSV methods, while a detailed technical description of the algorithms will be given in subsection 3.4.5.

The SSV1 method

We start with a brief sketch of the SSV1 method. The algorithm iterates over a growing test dimension $d = 1, 2, \ldots, d_{\max}$ and stops in case that either its decision rule accepts the current d as estimated intrinsic dimension (i.e., the output value), or d reaches $d_{\max}$. Consequently, we require $m < d_{\max} \leq D$. For growing d, we now consider local datasets $\mathcal{S}_i$ consisting of the point $\boldsymbol{x}_i$ and its k_d nearest neighbors, where $k_d \geq d + 2$. For each set $\mathcal{S}_i$, the radius r_i of its D-dimensional minimum bounding ball is computed, i.e., the ball in $\mathbb{R}^D$ containing all points in $\mathcal{S}_i$ with minimal radius. Next, a large number C of random d-simplices are sampled (with their $(d + 1)$ vertices drawn from $\mathcal{S}_i$) and their average volume $\overline{V}_i$ is calculated. The quotient of $\overline{V}_i$ and the volume of the d-dimensional ball[2] with radius r_i is then compared with the expected value $\nu(d)$. Thus, for the current test dimension d and each $i = 1, \ldots, N$, we consider the (random) quantities

$$q_i(d) = \overline{V}_i \cdot \left[\mathrm{vol}_d\left(B^d_{r_i}(\mathbf{0})\right) \cdot \nu(d)\right]^{-1} = \overline{V}_i \cdot \left[\xi(r_i, d, d)\right]^{-1}, \tag{3.102}$$

[2] *Mind the difference:* the radius r_i has been computed with respect to the D-dimensional minimum bounding ball; now we consider the d-dimensional ball of the same radius, where $d < D$. A justification is provided in the upcoming descriptions.

where $\nu(\cdot)$ and $\xi(\cdot,\cdot,\cdot)$ are defined as in eq. (3.98) and (3.100), respectively. Finally, the results from the local datasets $\mathcal{S}_i$ are combined into a global result — the exact proceeding is explained in the following. The algorithm stops once d is considered to be the valid dimension estimate or reaches its maximum $d_{\max}$.

Now the crucial question is: Which values of $q_i(d)$ do we expect in the three different cases $d < m$, $d = m$ and $d > m$? Recall that m is the true intrinsic dimension of the dataset and we assume that, if the points in $\mathcal{S}_i$ are enclosed in a sufficiently small ball, they are distributed according to the m-dimensional uniform distribution. Consequently and according to the definition of the constants $\nu(d)$, in the case $d = m$, we expect $q_i(d) = 1$. For the case of $d < m$, our intuition is that $q_i(d)$ should be larger than 1. To see this, let us first analyze the situation for $d = 1$ and growing $m \geq 2$. Here, note that $\nu(1) = \frac{1}{3}$ is the expected length of the line segment determined by two random points within the interval $[0, 1]$. For $m = 2$ or $m = 3$, respectively, we are looking for the expected length of the line segment determined by two random points from the uniform distribution in the sphere or the ball with volume 1, respectively. Intuitively, we would say that these values should be larger than $\nu(1)$, since the line segments can "spread out" in more dimensions.

In fact, theorem 3.7 gives us a straightforward way to compute the expectation of our random quantity $q_i(d)$ for $d \leq m$ as

$$\mathbb{E}\left[q_i(d)\right] = \frac{\xi(r_i, m, d)}{\xi(r_i, d, d)} = \frac{\xi(1, m, d)}{\xi(1, d, d)} =: f(m, d), \tag{3.103}$$

where $\xi(\cdot,\cdot,\cdot)$ is defined in eq. (3.100).

One can easily verify that for fixed d, $f(m, d)$ is monotonically increasing in m. Now, for our dimension estimation procedure, it is crucial to be able to distinguish between $q_i(m-1)$ and $q_i(m) = 1$ (in expectation). The corresponding numerical values of $f(m, d)$ for $d = m - 1$ and $d = m - 2$ can be found in table 3.4. Here, it can be seen that the expected values $f(m, m-1)$ considerably deviate from 1 and furthermore, the deviation gets slightly larger for higher intrinsic dimension m.

Summing up the theory so far, the computation of the quantities $q_i(d)$ for growing test dimension $d = 1, 2, \ldots, m$ should yield a monotonically decreasing series of values approaching 1 from above, where $q_i(\hat{d}) \approx 1$ is a good indicator for $\hat{d}$ being the true intrinsic dimension m.

m	2	3	4	5	6	7	8	9	10	20	30
$f(m, m-1)$	1.36	1.58	1.76	1.93	2.07	2.21	2.34	2.46	2.57	3.52	4.27
$f(m, m-2)$	–	1.54	1.96	2.35	2.73	3.10	3.48	3.85	4.22	7.90	11.6

Table 3.4: Numerical values (rounded to 3 digits) of $f(m, d)$ (see eq. (3.103)).

Finally, let us consider the case where $d > m$. Clearly, under our current assumption of noisefree data, and ignoring the local curvature of the manifold for now, any d-simplex with vertex points sampled from an m-dimensional uniform distribution will be degenerated and thus have a volume of zero. As a result, we also have $\mathbb{E}[q_i(d)] = 0$ for all $d > m$. Subsuming our considerations, the expectation for our empirical quantity is:

$$\mathbb{E}[q_i(d)] \begin{cases} > 1 & \text{for } d < m; \\ = 1 & \text{for } d = m; \\ = 0 & \text{for } d > m. \end{cases} \tag{3.104}$$

The last step of our method is the computation of the arithmetic mean

$$\overline{q}(d) = \frac{1}{N} \sum_{i=1}^{N} q_i(d) \tag{3.105}$$

and the final estimate is determined as

$$\hat{m} = \min_{d=1,\ldots,d_{\max}} \{d : \ \overline{q}(d+1) < 1 - \epsilon\} \tag{3.106}$$

for some tolerance parameter $0 < \epsilon \ll 1$.

In theory so far, the case-by-case analysis (3.104) qualifies our $q_i(d)$ as a well-suited criterion for the determination of the ID m. In practice however, a prudent choice of ϵ is very important, since for $d > m$ we rather expect the parameter's range as $0 \leq q_i(d) < 1$ due to the influence of noise and curvature of the manifold. In subsection 3.4.4, some experiments will in fact show that the presence of noise has a particularly large effect on the volumes of higher-dimensional simplices and thus, the presented SSV1 method can eventually lead to overestimation of the intrinsic dimension. This is one of the reasons the SSV2 method has been coined for.

Before we introduce the alternative approach, let us briefly discuss our preference for the use of bounding balls. First of all, the approach of a high-dimensional grid (or space-partitioning scheme) in the data space $\mathbb{R}^D$ does not seem appealing to us, because of well-known problems concerning complexity, potential sparse data, and also because of the simple fact that a d-simplex requires $(d+1)$ vertex points. Hence, the idea of examining nearest neighbor points and some adequate bounding body appears to be much more advisable. The most common variants of minimum bounding objects include the convex hull, D-dimensional balls, or boxes, which could be either axis-aligned or arbitrarily rotated in order to minimize their volume. Axis-aligned bounding boxes can be rapidly computed by trivial routines, but they are likely to include a large amount of empty space. Moreover, given some D-dimensional bounding box, there is no straightforward way to derive (the volume of) an appropriate d-dimensional bounding object (recall that $d \ll D$), and finally, the explicit expected volume values — as provided by theorems 3.6 and 3.7 for bounding balls — are not available. The latter two arguments are also

valid for rotated boxes as well as the convex hull. In contrast, balls have the beneficial property that, given D-dimensional data contained in some m-dimensional subspace, the radius of their D-dimensional minimum bounding ball coincides with the radius of their m-dimensional bounding ball within the m-dimensional subspace. Hence, the radius of the D-dimensional minimum bounding ball is a useful quantity and our proceeding described above is reasonable. Finally, the isotropic nature of the ball does not favor any direction which is another desirable property since the relevance of the different coordinate directions is completely unknown a priori.

The SSV2 method

The objective of the SSV2 method is to attenuate two drawbacks of the SSV1 method. The first one has already been mentioned — the presence of noise can lead to substantial overestimation of higher IDs. A second shortcoming is related with the runtime performance of the method. While many IDE methods, e.g. as the ones presented in subsection 3.2.2, are based on the computation of basic low-dimensional quantities, generally Euclidean distances or angles, the SSV1 method relies on the calculation of simplex volumes in dimensions up to the intrinsic dimension m of the data. Consequently, the asymptotic theoretical as well as practical runtime of its estimation routine, albeit not dominated by the ambient space dimension D, scales with the intrinsic dimension m and thus often exceeds the average runtime of other IDE methods.

The SSV2 method tackles these issues by evaluating only volumes of low-dimensional simplices of dimensions $s = 1, 2, \ldots, s_{\max}$ and comparing their averages with the associated expected values from Miles' theorem 3.7. This also allows for the estimation of ID values larger than $s_{\max}$. As one might expect, choosing a larger $s_{\max}$ entails both more precise estimation results and higher computation runtimes. Furthermore, while the SSV1 method analyzes only d-dimensional simplex volumes for some particular test dimension d, the SSV2 approach compares volumes of s-dimensional simplices, for *multiple* $s \leq d$, against their respective expected values. The current test dimension d is selected as the estimated ID if, for at least one of the values $s \leq d$, the average volume associated with s fulfills the condition for d-dimensional structures. The idea behind this strategy is that, while higher-dimensional simplex volumes might be influenced by noise to a higher degree, lower-dimensional simplices will still allow to detect the true underlying ID.

The SSV2 method features three consecutive stages. The first stage covers the cases where $\hat{m} < s^{(1)}_{\max}$, the second stage is deployed for $s^{(1)}_{\max} \leq \hat{m} < s^{(2)}_{\max} = s_{\max}$, and the third stage finally deals with all cases where $s^{(2)}_{\max} \leq \hat{m} \leq d_{\max}$. In our actual implementation, the constants are fixed as $s^{(1)}_{\max} = 5$ and $s^{(2)}_{\max} = 10$. The purpose of these different stages is increased computational efficiency, which will become clear in the upcoming explanations.

Before we describe each stage in more detail, we first illuminate the general functionality of the SSV2 approach. Similarly to the SSV1 approach, for growing test dimension

$d = 1, 2, \ldots, d_{\max} \leq D$, s-dimensional simplex volumes with $s \leq d$ are analyzed and the algorithm is terminated with output value $\hat{m} = d$ as soon as the current variable d is considered to be the correct estimate. As before, let $\mathcal{S}_i$ denote the set consisting of the point $\boldsymbol{x}_i$ and its k nearest neighbors, let further r_i be the radius of the D-dimensional minimum bounding ball of $\mathcal{S}_i$. Next, $\overline{V}_i(s)$ denotes the (empirical) average volume of s-dimensional simplices with their $(s+1)$ vertices randomly drawn from $\mathcal{S}_i$. The crucial local quantities are now given by

$$q_i(s, d) = \overline{V}_i(s) \cdot [\xi(r_i, d, s)]^{-1}, \tag{3.107}$$

where $\xi(\cdot, \cdot, \cdot)$ is defined as in eq. (3.100).

While in each of the three stages, a slightly different proceeding applies, the common decision criterion for choosing the ID value $\hat{m} = d - 1$ is given by

$$\overline{q}(s, d) < 1 - \epsilon, \tag{3.108}$$

where $\overline{q}(s, d) = \sum_{i=1}^{N} q_i(s, d)$ represents the average of the local quantities $q_i(s, d)$ and $\epsilon > 0$ is a tolerance parameter.

Let us now focus on the detailed estimation process. To begin with, note that — in contrast to the SSV1 method — the SSV2 method works with two fixed numbers of nearest neighbors, k_1 and k_2. Since the maximum dimension of considered simplices is fixed here, there is no need of growing numbers of nearest neighbors. In theory, a single constant number of NNs would be enough. On the other hand, the use of two different NN sizes allows a better estimation of IDs $\hat{m} \geq s_{\max}^{(2)}$. In our implementation, the values are chosen as $k_1 = 12$ and $k_2 = 30$. In short, stage I works with only k_1 nearest neighbors, since a small number of NNs is sufficient here and speeds up the computations. Stage II also relies on the estimates based on k_1 nearest neighbors, nevertheless, it determines k_2 NNs and computes two potentially different local results for each point, for both k_1 and k_2. However, these two results are only combined in case stage III is required to estimate intrinsic dimensions $\hat{m} \geq s_{\max}^{(2)}$. Of course, it would be more intuitive to compute the estimates for k_2 only in stage III, where they are actually used. This would accelerate stage II, but also decelerate stage III to a greater extent, since the whole process of NN searching and random vertex sampling had to be accomplished once more. Hence, we opted for the combined computation in the second stage.

The precise proceeding is now as follows. **Stage I** is based on local datasets $\mathcal{S}_i$ defined by k_1 nearest neighbors. In a first step, the local average simplex volumes $\overline{V}_i(s)$ are evaluated for all $s = 1, \ldots, s_{\max}^{(1)}$. Next, for growing test dimension $d = 1, \ldots, s_{\max}^{(1)}$, the local quantities (3.107) are computed for all $s = 1, \ldots, d$. If, for any of those $s = 1, \ldots, d$, the criterion (3.108) is fulfilled, the procedure is terminated with output $\hat{m} = d - 1$. The sole exception is the case $s = 1$ (1-dimensional simplices, i.e., Euclidean distances), which is considered only for test dimensions $d = 1, 2$. The reason for this decision emerged from our experimental results; here, for fixed $d > 2$, the quantity $\overline{q}(1, d)$ often

featured a behavior differing from the other $\overline{q}(s,d)$ for $s \geq 2$.

If no intrinsic dimension $\hat{m} < s_{\max}^{(1)}$ has been found, the algorithm initiates **stage II**. Here, the first step again consists of the computation of local average simplex volumes, where the simplex dimension now ranges from $s = s_{\max}^{(1)} + 1$ to $s = s_{\max}^{(2)}$. Besides, the complete procedure of volume computation is performed two times, for both numbers of nearest neighbors k_1 and k_2, thus yielding the average volumes $\overline{V}_i^{(z)}(s)$ for $z = 1, 2$, where z indicates the use of either k_1 or k_2. However, in stage II, only the results for k_1 are utilized since they generally provide the better estimates here; the results for k_2 are pre-computed for stage III. Finally, for growing test dimension $d = s_{\max}^{(1)} + 1, \ldots, s_{\max}^{(2)}$, the quantities $q_i(s,d)$ (see (3.107), where $\overline{V}_i(s) = \overline{V}_i^{(1)}(s)$) are evaluated for each $s = s_{\max}^{(1)} + 1, \ldots, d$, and the decision criterion (3.108) applies as before.

Stage III finally deals with the case where $\hat{m} \geq s_{\max}^{(2)}$. Now, for $z = 1, 2$, and for test dimensions $d = s_{\max}^{(2)} + 1, \ldots, d_{\max}$, the quantities

$$q_i^{(z)}(s,d) = \overline{V}_i^{(z)}(s) \cdot \left[\xi\left(r_i^{(z)}, d, s\right)\right]^{-1}, \qquad z = 1, 2 \tag{3.109}$$

are computed for each $s = s_{\max}^{(1)} + 1, \ldots, s_{\max}^{(2)}$. This proceeding leads to two (potentially different) estimates $\hat{m}^{(z)}$, again based on the criterion (3.108), i.e., $\hat{m}^{(z)} = d - 1$, if $\overline{q}^{(z)}(s,d) < 1 - \epsilon$ for any $s = s_{\max}^{(1)} + 1, \ldots, s_{\max}^{(2)}$. In the final step, the two estimates $\hat{m}^{(1)}$ and $\hat{m}^{(2)}$ are combined as follows. If both estimates have been assigned a value less than $d_{\max}$, the final estimate is defined as

$$\hat{m} = \begin{cases} \hat{m}^{(1)} & \text{if } \hat{m}^{(1)} \leq k_1, \\ \frac{1}{2} \cdot \left(\hat{m}^{(1)} + \hat{m}^{(2)}\right) & \text{if } \hat{m}^{(1)} > k_1 \ \wedge \ \hat{m}^{(2)} \leq k_2, \\ \hat{m}^{(2)} & \text{otherwise.} \end{cases} \tag{3.110}$$

In case that only one of the estimates is below $d_{\max}$, the output $\hat{m}$ of course gets this value, and $d_{\max}$ otherwise.

The reason for combining the two estimates $\hat{m}^{(1)}$ and $\hat{m}^{(2)}$ as in (3.110) is quite simple: a larger number of nearest neighbors is more suitable to capture the local structures of high-dimensional data. In our early experiments, we in fact evaluated another variant of the SSV2 method with three different numbers of NNs k_1, k_2, k_3 leading to even more precise estimates for some datasets with high ID. While the advantages of this multiscale approach are undeniable, on the downside, both the computation of many NN points and of their associated minimum bounding ball get increasingly expensive in higher-dimensional spaces. On top of that, there is no straightforward way of combining three or more estimates in our setting. Therefore, we found the use of two different numbers of nearest neighbors to be a good tradeoff.

Finally, let us examine the characteristics of the crucial quantity (3.107), which is the empirical average simplex volume divided by the corresponding expected value. Analogous considerations as for the SSV1 method lead to the conclusion that, given some

intrinsically m-dimensional data, the expectation of these quantities is given by

$$\mathbb{E}\left[q_i(s,d)\right] = \frac{\xi(r_i,m,s)}{\xi(r_i,d,s)} = \frac{\xi(1,m,s)}{\xi(1,d,s)} =: g(s,m,d). \tag{3.111}$$

Similarly as above, we now have

$$\mathbb{E}\left[q_i(s,d)\right] \begin{cases} > 1 & \text{for } d < m \\ = 1 & \text{for } d = m \\ < 1 & \text{for } d > m \end{cases} \forall s \leq \min\left\{d,m\right\}. \tag{3.112}$$

m	2	3	4	5	6	7	8	9	10
$g(1,m,m-1)$	1.358	1.136	1.073	1.046	1.032	1.023	1.018	1.014	1.011
$g(2,m,m-1)$	–	1.582	1.240	1.136	1.088	1.062	1.047	1.036	1.029
$g(3,m,m-1)$	–	–	1.765	1.332	1.195	1.130	1.094	1.071	1.056

m	9	10	11	12	13	14	15	16	17
$g(8,m,m-1)$	2.459	1.715	1.459	1.329	1.251	1.199	1.163	1.136	1.116
$g(9,m,m-1)$	–	2.574	1.782	1.507	1.366	1.281	1.225	1.185	1.155
$g(10,m,m-1)$	–	–	2.685	1.846	1.553	1.403	1.311	1.250	1.206

Table 3.5: Numerical values (rounded to 4 digits) of $g(s,m,d)$ (see eq. (3.111)) with the particular test dimension $d = m-1$ and simplex dimension $s = 1,2,3$ *(above)* and $s = 8,9,10$ *(below)*.

Table 3.5 shows some numerical values of $g(s,m,d)$ for test dimension $d = m-1$ for $s = 1,2,3$ and $s = 8,9,10$, respectively. Of course, the values for $d = m-1$ are crucial for an efficient estimation procedure, since we have to distinguish between the case $d = m-1$ and $d = m$. From Table 3.5, we realize that low-dimensional simplices ($s = 1,2,3$) are not very well-suited to estimate IDs of $m = 8$ and above, since $g(s,m,m-1)$ tends to 1 relatively fast. This tendency is not as severe for higher-dimensional simplices ($s = 8,9,10$), even though also here — as expected — the estimation of IDs m which are significantly larger than s becomes increasingly difficult.

Before we unveil the exhaustive algorithmic description of our two SSV methods, we would like to present an empirical evaluation of the influence of noise on higher-dimensional simplex volumes.

3.4.4 Influence of noise on simplex volumes

To substantiate our conjecture that volumes of high-dimensional simplices are affected to a higher degree by noise than volumes of low-dimensional simplices, we evaluate the following test cases. We randomly sample 100000 data points from an m-dimensional unit ball and embed those points into $\mathbb{R}^D$ using a random rotation. In the first setting, we choose $m = 4$ and $D = 4, 10, 20, 50$, in the second setting, we have $m = 10$ and $D = 10, 20, 50, 100, 500$. The data are now perturbed with D-dimensional Gaussian noise of some standard deviation σ. Next, for each $s = 1, 2, \ldots, m$, we calculate the average value $\overline{q}(s, m)$ of 20000 random s-simplex volumes (with vertices sampled from the data points) and consider the quotient

$$f(s, D, \sigma) := \frac{\overline{q}(s, m)}{\xi(1, m, s)}, \tag{3.113}$$

where $\xi(\cdot, \cdot, \cdot)$ is defined by eq. (3.100).

The experiment is first performed for the noisefree case (denoted by $\sigma = 0$), where the result should be close to 1 in each case, and then for Gaussian noise of $\sigma_1 = 0.01$ and $\sigma_2 = 0.05$. Tables 3.6 and 3.7 show the associated empirical results for ID $m = 4$ and $m = 10$, respectively. Each test case has been evaluated 20 times for different sampled data and the averages are presented here.

Let us now consider the experimental results. For the noisefree case, we see that the values are close to 1 which confirms that the volume computation is stable and the number of 20000 random simplices is sufficient for this experiment.

In the first setting, we note that Gaussian noise with $\sigma = 0.01$ only leads to moderate deviation of the measured volumes from the predicted values. For $\sigma = 0.05$, we identify two clear trends. First and very naturally, a higher embedding dimension D leads to a larger deviation. However for fixed D, the deviation quickly gets larger with growing s, i.e., for higher-dimensional simplex volumes. Note in particular the leap from $s = 3$ to $s = 4$.

The same tendencies can again be found in the second setting. Especially for higher noise level $\sigma = 0.05$ and ambient dimension $D \geq 50$, the deviation of s-dimensional simplex volumes is disproportionately large for s approaching the intrinsic dimension $m = 10$ as compared to $s = 1, 2$.

This effect suggests that the SSV1 method is likely to overestimate the intrinsic dimension of noisy datasets, at least for higher levels of Gaussian noise and large ambient dimension D.

A theoretical analysis of the influence of noise on expected average simplex volumes of different dimensionalities would certainly be very enlightening for our purposes. While we assume this to be a challenging problem, we recommend this topic for future research.

s	1	2	3	4	s	1	2	3	4
$f(s,4,0)$	0.999	1.000	1.000	1.000	$f(s,20,0)$	1.000	1.001	0.999	0.998
$f(s,4,0.01)$	1.000	1.000	1.001	1.001	$f(s,20,0.01)$	1.002	1.005	1.012	1.055
$f(s,4,0.05)$	1.007	1.014	1.022	1.027	$f(s,20,0.05)$	1.042	1.113	1.273	1.834
$f(s,10,0)$	1.000	1.000	0.999	0.999	$f(s,50,0)$	1.001	0.999	1.000	0.998
$f(s,10,0.01)$	1.001	1.002	1.005	1.023	$f(s,50,0.01)$	1.005	1.012	1.033	1.133
$f(s,10,0.05)$	1.020	1.052	1.120	1.373	$f(s,50,0.05)$	1.105	1.289	1.701	3.113

Table 3.6: Numerical values of $f(s, D, \sigma)$ (see eq. (3.113)) for fixed ID $m = 4$ and ambient dimensions $D = 4, 10, 20, 50$.

s	1	2	3	4	5	6	7	8	9	10
$f(s,10,0)$	1.000	1.000	1.000	1.000	0.999	0.999	0.999	0.999	0.998	0.997
$f(s,10,0.01)$	1.001	1.001	1.001	1.002	1.001	1.002	1.002	1.002	1.002	1.002
$f(s,10,0.05)$	1.014	1.028	1.042	1.056	1.070	1.085	1.101	1.115	1.130	1.148
$f(s,20,0)$	1.000	1.000	1.000	0.999	0.999	0.999	0.998	0.998	0.999	0.999
$f(s,20,0.01)$	1.001	1.002	1.004	1.006	1.008	1.012	1.016	1.025	1.048	1.154
$f(s,20,0.05)$	1.030	1.064	1.103	1.152	1.215	1.305	1.442	1.680	2.193	3.878
$f(s,50,0)$	1.000	1.000	1.000	1.000	0.999	0.999	0.998	0.999	1.000	1.003
$f(s,50,0.01)$	1.003	1.006	1.011	1.017	1.025	1.037	1.056	1.092	1.178	1.500
$f(s,50,0.05)$	1.076	1.169	1.290	1.454	1.690	2.051	2.659	3.820	6.565	16.66
$f(s,100,0)$	1.000	1.000	1.000	1.000	0.999	1.000	1.000	1.000	1.000	0.999
$f(s,100,0.01)$	1.006	1.014	1.023	1.035	1.053	1.078	1.119	1.196	1.373	1.998
$f(s,100,0.05)$	1.147	1.340	1.609	2.002	2.612	3.630	5.505	9.490	20.11	64.71
$f(s,500,0)$	1.000	1.000	1.000	1.000	1.000	1.001	1.001	1.001	1.001	1.000
$f(s,500,0.01)$	1.031	1.070	1.120	1.188	1.285	1.431	1.670	2.115	3.126	6.642
$f(s,500,0.05)$	1.602	2.670	4.689	8.764	17.65	38.93	96.44	279.4	1023	5848

Table 3.7: Numerical values of $f(s, D, \sigma)$ (see eq. (3.113)) for fixed ID $m = 10$ and ambient dimensions $D = 10, 20, 50, 100, 500$.

3.4.5 Algorithmic description of the SSV1 and SSV2 method

In this subsection, we give a complete algorithmic description of our two SSV methods. The important issue of the efficient computation of simplex volumes has already been discussed in detail in section 3.3. Next, we present two more algorithmic components that are employed likewise in both variants. Subsequently, we introduce the details and the pseudocode for each approach individually.

Auxiliary tree structure – preprocessing of the data

The rapid determination of a fixed number of nearest neighbor (NN) points noticeably reduces the overall computation time of our algorithm. For this reason, in a preprocessing step, our method establishes an auxiliary data structure for the entire dataset. The k-NN search problem in high dimensions, where the dissimilarity measure is a distance metric, more specifically in our context the Euclidean distance, is a highly non-trivial task and has led to a variety of approaches, both for exact and approximate NN search, see e.g. [SDI06], [PM05], [ZADB06]. Several competing branches of techniques exist, such as index trees, hashing, vector quantization, particular graph algorithms, just to mention a few.

We rely on a well-established, simple technique for two reasons. First, the focus of our work is the correct estimation of the intrinsic dimension, whereas speed is only a secondary issue. Furthermore, most of the datasets we deal with will be of rather low (< 10) intrinsic dimension. In [VKD09], the authors examine the adaptivity of different tree structures to the intrinsic dimensionality of the data. Even though axis-parallel splitting rules can not adapt well for special datasets, it is shown that on average, a certain variant of the *kd*-tree performs qualitatively similar to more complex structures like the *random projection tree*, the *principal direction tree*, and the *two means tree*.

Consequently, we utilize this particular variant of the *kd*-tree in our method. In a nutshell, a *kd*-tree is a binary tree, where each inner node represents a hyperplane dividing the space into two half-spaces. The left and right subtree of that node hold the corresponding points, left and right of the hyperplane. Furthermore, each such hyperplane is orthogonal to one of the coordinates axes. Different selections of those coordinate axes lead to different variants of *kd*-trees. In our case, in each partitioning step, the splitting direction is selected as the coordinate axis with maximum spread, and the median along that direction defines the corresponding split position.

As mentioned above, the only purpose of the tree structure in our method is the acceleration of the NN search process. The use of different data structures does not affect the algorithmic output in any way. Naturally, more complex approaches, e.g. as introduced in [Vai89], [SSV07], could be used in order to optimize the speed-up for the all-nearest-neighbor search problem. However, in order to keep our method simple and easily replicable, we choose to work with the *kd*-tree described above.

Minimum bounding ball

The computation of the unique D-dimensional minimum bounding ball (also named smallest enclosing ball / sphere / disk) for a given point set $\mathcal{S} \subset \mathbb{R}^D$ of size $|\mathcal{S}| = n$ is a non-trivial task for which multiple different approaches exist.

For our purposes, we tested and compared two recent methods. The first method has been introduced in [FGK03], computes the exact solution, is one of the fastest codes available for lower dimensions and still able to handle point sets in dimensions of up to several thousands efficiently. The second approach presented in [LK13] computes an approximate solution, is very easy to implement and can be considerably faster than its competitors for very high-dimensional data.

The first approach [FGK03] by Fischer, Gärtner and Kutz (code available at [FGK15]) is an enhancement of the algorithm presented by Gärtner in [Gär99], which is itself a carefully optimized and numerically stable variant of an early approach by Welzl in [Wel91]. All three methods have expected linear runtime in the number of points n. They iteratively modify the current bounding ball (until it represents the actual solution) via computing new support sets of at most $(D + 1)$ points on the boundary of the current ball. Even though the worst-case runtime complexity is exponential in D, the performance of the most recent method is very good for low-dimensional problems but suffers e.g. for almost cospherical point sets in very high dimensions.

The second approach [LK13] by Larsson and Källberg is an approximation algorithm: for a specified precision $\delta > 0$, a bounding ball for $\mathcal{S}$ with radius $r \leq (1 + \delta)r^*$ is computed, where r^* shall denote the radius of the minimum bounding ball. The best time complexity for this problem achieved until today is $\mathcal{O}(n \cdot D/\delta)$, an example is the algorithm presented in [Yil08], which also provides a solid survey on similar methods. The method in [LK13] of our choice comes with a runtime bound of $\mathcal{O}(n \cdot D/\delta + D/\delta^3)$. However, the authors still claim the performance-wise superiority of their method for values of $\delta \geq 0.001$, and in fact, the second summand in the complexity is due to an additional optimization routine reducing the actual runtime in many practical cases.

Our scenario is somewhat different from most experiments in the literature, since we are usually interested in minimum bounding balls for small sets of size $n \ll 100$, where the ambient dimension D can however be quite large. For this reason, we examined both methods embedded in the environment of our main SSV algorithms and compared their respective performance. While for smaller dimensions of $D < 50$ we found no measurable differences, for higher-dimensional point sets the approximation algorithm (with $\delta = 0.001$) performed significantly faster. Since the final estimation results are not influenced by the fact that the bounding ball radius is only δ-exact, we ultimately opted for the approach by [LK13] with fixed approximation parameter $\delta = 0.001$.

SSV1 – Input, output and constants

The mandatory input is the data dimension D, the number of data points N, and the data $\mathcal{X} = \{\boldsymbol{x}_{i=1,\ldots,N}\}$. The (optional) input parameter $d_{\max}$ provides an upper bound

for the output, i.e., the estimated intrinsic dimension; its default value is of course $d_{\max} = D$. The data $\{\boldsymbol{x}_{i=1,\dots,N}\}$ is supposed to be intrinsically low-dimensional with intrinsic dimension m as described in subsection 3.4.2. The only output parameter is the estimated intrinsic dimension $\hat{m}$. Constants used in the algorithm are C, the number of test simplices considered for each local dataset, further $k_{\min}$, the minimal number of nearest neighbor points forming the local datasets, and ϵ, a tolerance parameter which is mandatory due to the influence of manifold curvature and noise in the data. Since the method examines random simplices with vertices drawn from relatively small local subsets of the data, we found that $C = 1000$ is a sufficiently large number to compute reliable values of the local average volumes. The number of nearest neighbors is adapted to the current test dimension. Still, the minimum number of NNs must not be to small, which is why we choose $k_{\min} = 12$ in our experiments. Finally, the tolerance parameter is fixed as $\epsilon = 0.05$. This relatively small value allows precise ID estimations of (noisefree) high-dimensional data, while, on the other hand, the ID of noisy data is likely to be overestimated in certain cases.

SSV1 – Dimension estimation procedure

The algorithmic sketch of the SSV1 method is presented in figure 3.1.

First, the kd-tree structure $\mathbb{T}_{\mathcal{X}}$ is built which will accelerate the nearest neighbor search process to construct the local sets $\mathcal{S}_i$. The output variable $\hat{m}$ is initialized as D, the standard value if no lower ID can be found.

The outer FOR-loop marches through all possible values $d = 1, \dots, d_{\max}$ of the intrinsic dimension and is aborted immediately, if $(d-1)$ is considered to be the correct output value $\hat{m}$. For fixed d, the algorithm analyzes volumes of random d-simplices with $(d+1)$ vertices. The number of NN points is chosen as $k_d = \max(d+3, k_{\min})$. This guarantees a sufficiently large number of points to draw the $(d+1)$ vertices from.

The intermediate FOR-loop marches through all local datasets, which are considered separately. For the purpose of reproducibility, our method passes through *all* data points $\boldsymbol{x}_i$ and computes the corresponding nearest neighbor set $\mathcal{S}_i = \left\{ \boldsymbol{x}_i^{(j)} \;\middle|\; j = 0, \dots, k_d \right\}$ using the tree structure $\mathbb{T}_{\mathcal{X}}$.

For a fixed set $\mathcal{S}_i$, in the next step, the approximate minimum (D-dimensional) bounding ball, denoted by $B(\mathcal{S}_i)$, with its radius r_i is calculated. This step is accomplished via the algorithm presented in [LK13] as described above, where the resulting radius is δ-exact with $\delta = 0.001$.

In the inner FOR-loop, random subsets of $(d+1)$ points are drawn from $\mathcal{S}_i$ resulting in random sample simplices. The average volume $\overline{V}_i$ of C of those simplices is computed as explained above.

Subsequently, $\overline{V}_i$ is divided by the volume of the d-dimensional ball with radius r_i, and the result is again divided by the corresponding expected volume $\nu(d)$ to yield the local quantities

$$q_i(d) = \overline{V}_i \cdot \left[\mathrm{vol}_d(B_{r_i}^d(\mathbf{0})) \cdot \nu(d) \right]^{-1}, \tag{3.114}$$

and finally, their arithmetic mean:

$$\overline{q}(d) = \frac{1}{N} \sum_{i=1}^{N} q_i(d). \tag{3.115}$$

As explained in subsection 3.4.3, we expect the following results: $\overline{q}(d) > 1$ for $d < m$, $\overline{q}(d) \approx 1$ for $d = m$, and $0 \leq \overline{q}(d) < 1$ for $d > m$. Thus, the EXIT-condition for the outer loop is defined as $q(d) < 1 - \epsilon$, where the tolerance parameter ϵ is required due to the effects of manifold curvature and noise in the data.

A further remark shall be given with regard to the averaging in (3.115). In a setting

---

SSV1$(D, \boldsymbol{x}_{i=1,\dots,N}, d_{\max})$
Input: D = dimension; $\boldsymbol{x}_1, \dots, \boldsymbol{x}_N$ = data; $d_{\max}$ = max. intrinsic dimension
Output: $\hat{m}$ = estimated intrinsic dimension
Constants: $k_{\min}$ = min. no. of NNs; C = no. of test simplices; ϵ = tolerance parameter

---

$\mathbb{T}_{\mathcal{X}} \leftarrow$ **ConstructTree** $(\{\boldsymbol{x}_{i=1,\dots,N}\})$
$\hat{m} \leftarrow D$
for $d = 1, \dots, d_{\max}$ **do**
  $k_d \leftarrow \max\{d + 3, k_{\min}\}$
  $\overline{q} \leftarrow 0$
  for $i = 1, \dots, N$ **do**
    $\mathcal{S}_i \leftarrow \left\{ \boldsymbol{x}_i^{(j)} \,\middle|\, j = 0, \dots, k_d \right\}$ /* local NN dataset */
    $r_i \leftarrow \text{radius}(B(\mathcal{S}_i))$
    $\overline{V}_i \leftarrow 0$
    for $t = 1, \dots, C$ **do**
      $\Delta_t \leftarrow d$-simplex with vertices randomly drawn from $\mathcal{S}_i$
      $\overline{V}_i \leftarrow \overline{V}_i + \text{vol}(\Delta_t)$
    $\overline{V}_i \leftarrow \overline{V}_i / C$
    $\overline{q} \leftarrow \overline{q} + \overline{V}_i \cdot \left[\text{vol}\left(B_{r_i}^d(\mathbf{0}) \right) \cdot \nu(d) \right]^{-1}$
  $\overline{q} \leftarrow \overline{q} / N$
  if $\overline{q} < 1 - \epsilon$ **then**
    $\hat{m} \leftarrow d - 1$
    break /* ID found; exit outer for-loop */
return $\hat{m}$

Figure 3.1: Basic structure of the SSV1 dimension estimation algorithm. Note that $\nu(d)$ are the constant values defined in (3.98) denoting the expected random simplex volume within a d-dimensional ball with volume 1.

with data sampled from some particular non-uniform probability distribution it can be beneficial to assign different weights to different subsets, for example according to their particular bounding ball radii. In our setting — for a uniform distribution — there is no reason to prefer some particular weighting to the unweighted average.

SSV2 – Input, output and constants

The input parameters D, N, $\mathcal{X} = \{\boldsymbol{x}_{i=1,\dots,N}\}$, and $d_{\max}$, as well as the only output parameter $\hat{m}$ are the same as for the SSV1 method. When it comes to the constants, C (with default value $C = 1000$) still denotes the number of test simplices for each local set $\mathcal{S}_i$. The tolerance parameter is $\epsilon = 0.05$ as in the SSV1 approach.

The more important constants are $s_{\max}^{(1)}$ and $s_{\max}^{(2)}$, the maximal dimension of test simplices for stage I and II of the algorithm. As explained before, lower values can be used to speed up the computation, while higher values allow for a more precise estimation of datasets of high intrinsic dimensionality. Our choice is $s_{\max}^{(1)} = 5$ and $s_{\max}^{(2)} = 10$, which seems to be a good trade-off between precision and performance. Next, in stage II and III we use two different numbers k_1, k_2 of nearest neighbor points in order to yield more reliable estimates, again for data with high ID. Note that while the SSV1 method works with NN sets of variable size $|\mathcal{S}_i| = \max\{d+3, k_{\min}\}$ for test dimension d, the SSV2 method only utilizes these two fixed numbers of nearest neighbor points for each local subset. We choose $k_1 = 12$ equal to the parameter $k_{\min} = 12$ of the SSV1 method in order to obtain similar estimates of both methods for low IDs; finally, we found that the choice of $k_2 = 30$ leads to decent results for a wide spectrum of different datasets.

SSV2 – Dimension estimation procedure

The algorithmic structure of the SSV2 method is presented in figure 3.2 with the description of stage II and stage III given in figures 3.3 and 3.4, respectively.

To begin with, the kd-tree structure $\mathbb{T}_{\mathcal{X}}$ is built (accelerating the computation of the local sets $\mathcal{S}_i$), and the output variable $\hat{m}$ is initialized with its standard value D.

Next, recall that the local quantities are given by

$$q_i^{(z)}(s,d) = \overline{V}_i^{(z)}(s) \cdot \left[\xi\left(r_i^{(z)}, d, s\right)\right]^{-1}. \tag{3.116}$$

Here, the index $z \in \{1,2\}$ specifies the current number of NN points k_z, and $r_i^{(z)}$ is the radius of the (approximate) minimum bounding ball of the local NN set $\mathcal{S}_i^{(z)} = \left\{\boldsymbol{x}_i^{(j)} \,\middle|\, j = 0,\dots,k_z\right\}$. Correspondingly, $\overline{V}_i^{(z)}(s)$ denotes the (empirical) average volume of s-dimensional simplices with their $(s+1)$ vertices randomly drawn from $\mathcal{S}_i^{(z)}$. The decision criterion is given by

$$\overline{q}^{(z)}(s,d) < 1 - \epsilon, \tag{3.117}$$

where $\overline{q}^{(z)}(s,d) = \frac{1}{N}\sum_{i=1}^{N} q_i^{(z)}(s,d)$.

SSV2$(D, \boldsymbol{x}_{i=1,\dots,N}, d_{\max})$

Input: D = dimension; $\boldsymbol{x}_1, \dots, \boldsymbol{x}_N$ = data; $d_{\max}$ = max. intrinsic dimension
Output: $\hat{m}$ = estimated intrinsic dimension
Constants: $s_{\max}^{(i)}$ = max. dim. of simplices for stage $i = 1, 2$; k_1, k_2 = different no. of NNs; C = no. of test simplices; ϵ = tolerance parameter

1: $\mathbb{T}_{\mathcal{X}} \leftarrow$ **ConstructTree** $(\{\boldsymbol{x}_{i=1,\dots,N}\})$
2: $\hat{m} \leftarrow D$
3: /* STAGE I */
4: **for** $i = 1, \dots, N$ **do**
5: $\quad \mathcal{S}_i \leftarrow \left\{ \boldsymbol{x}_i^{(j)} \,\middle|\, j = 0, \dots, k_1 \right\}$ /* local NN dataset */
6: $\quad r_i \leftarrow \text{radius}(B(\mathcal{S}_i))$
7: $\quad$ **for** $s = 1, \dots, s_{\max}^{(1)}$ **do**
8: $\quad\quad \overline{V}_{i,s} \leftarrow 0$
9: $\quad\quad$ **for** $t = 1, \dots, C$ **do**
10: $\quad\quad\quad \Delta_t \leftarrow s$-simplex with vertices randomly drawn from $\mathcal{S}_i$
11: $\quad\quad\quad \overline{V}_{i,s} \leftarrow \overline{V}_{i,s} + \text{vol}\,(\Delta_t)$
12: $\quad\quad \overline{V}_{i,s} \leftarrow \overline{V}_{i,s}/C$
13: **for** $d = 1, \dots, s_{\max}^{(1)}$ **do**
14: $\quad$ **for** $s = 1, \dots, d$ **do**
15: $\quad\quad \overline{q}_{s,d} \leftarrow 0$
16: $\quad\quad$ **for** $i = 1, \dots, N$ **do**
17: $\quad\quad\quad \overline{q}_{s,d} \leftarrow \overline{q}_{s,d} + \overline{V}_{i,s} \cdot [\xi(r_i, d, s)]^{-1}$
18: $\quad\quad \overline{q}_{s,d} \leftarrow \overline{q}_{s,d}/N$
19: $\quad\quad$ **if** $\overline{q}_{s,d} < 1 - \epsilon$ **and** $(d \leq 2$ **or** $s \geq 2)$ **then**
20: $\quad\quad\quad \hat{m} \leftarrow \max\{d - 1, 1\}$
21: $\quad\quad\quad$ **goto** line 22 /* ID found; exit for-loop */
22: **if** $\hat{m} < D$ **then**
23: $\quad$ **return** $\hat{m}$
24: **else**
25: $\quad$ **SSV2StageII**$(D, \boldsymbol{x}_{i=1,\dots,N}, d_{\max})$

Figure 3.2: Basic structure of the SSV2 dimension estimation algorithm. Note that $\xi(r_i, d, s)$ are the expected values defined in eq. (3.100). Stage II and III are presented in figures 3.3 and 3.4, respectively.

SSV2StageII$(D, \boldsymbol{x}_{i=1,\ldots,N}, d_{\max})$: input, output, constants as in **SSV2**

1: **for** $z = 1, 2$ **do**
2: **for** $i = 1, \ldots, N$ **do**
3: $\mathcal{S}_i^{(z)} \leftarrow \left\{ \boldsymbol{x}_i^{(j)} \,\middle|\, j = 0, \ldots, k_z \right\}$ /* local NN dataset */
4: $r_i^{(z)} \leftarrow \text{radius}(B(\mathcal{S}_i^{(z)}))$
5: **for** $s = s_{\max}^{(1)} + 1, \ldots, s_{\max}^{(2)}$ **do**
6: $\overline{V}_{i,s}^{(z)} \leftarrow 0$
7: **for** $t = 1, \ldots, C$ **do**
8: $\Delta_t \leftarrow s$-simplex with vertices randomly drawn from $\mathcal{S}_i^{(z)}$
9: $\overline{V}_{i,s}^{(z)} \leftarrow \overline{V}_{i,s}^{(z)} + \text{vol}\,(\Delta_t)$
10: $\overline{V}_{i,s}^{(z)} \leftarrow \overline{V}_{i,s}^{(z)} / C$
11: **for** $d = s_{\max}^{(1)} + 1, \ldots, s_{\max}^{(2)}$ **do**
12: **for** $s = 2, \ldots, d$ **do**
13: $\overline{q}_{s,d} \leftarrow 0$
14: **for** $i = 1, \ldots, N$ **do**
15: $\overline{q}_{s,d} \leftarrow \overline{q}_{s,d} + \overline{V}_{i,s}^{(1)} \cdot \left[\xi \left(r_i^{(1)}, d, s \right) \right]^{-1}$
16: $\overline{q}_{s,d} \leftarrow \overline{q}_{s,d} / N$
17: **if** $\overline{q}_{s,d} < 1 - \epsilon$ **then**
18: $\hat{m} \leftarrow d - 1$
19: **goto** line 20 /* ID found; exit for-loop */
20: **if** $\hat{m} < D$ **then**
21: **return** $\hat{m}$
22: **else**
23: **SSV2StageIII**$(D, \boldsymbol{x}_{i=1,\ldots,N}, d_{\max})$

Figure 3.3: Stage II of the SSV2 algorithm.

SSV2StageIII$(D, \boldsymbol{x}_{i=1,\ldots,N}, d_{\max})$: input, output, constants as in **SSV2**

for $z = 1, 2$ **do**
  $\hat{m}^{(z)} \leftarrow D$
  for $d = s^{(2)}_{\max} + 1, \ldots, d_{\max}$ **do**
    for $s = s^{(1)}_{\max} + 1, \ldots, s^{(2)}_{\max}$ **do**
      $\overline{q}^{(z)}_{s,d} \leftarrow 0$
      for $i = 1, \ldots, N$ **do**
        $\overline{q}^{(z)}_{s,d} \leftarrow \overline{q}^{(z)}_{s,d} + \overline{V}^{(z)}_{i,s} \cdot \left[\xi\left(r^{(z)}_i, d, s\right)\right]^{-1}$
      $\overline{q}^{(z)}_{s,d} \leftarrow \overline{q}^{(z)}_{s,d}/N$
      if $\overline{q}^{(z)}_{s,d} < 1 - \epsilon$ **then**
        $\hat{m}^{(z)} \leftarrow d - 1$
        break; break /* exit both for(s)- and for(d)-loop */
if $\hat{m}^{(1)} < D$ **and** $\hat{m}^{(2)} < D$ **then**
  if $\hat{m}^{(1)} \leq k_1$ **then**
    $\hat{m} \leftarrow \hat{m}^{(1)}$
  else if $\hat{m}^{(2)} \leq k_2$ **then**
    $\hat{m} \leftarrow \text{round}\left(\left[\hat{m}^{(1)} + \hat{m}^{(2)}\right]/2\right)$
  else
    $\hat{m} \leftarrow \hat{m}^{(2)}$
else if $\hat{m}^{(2)} < D$ **then**
  $\hat{m} \leftarrow \hat{m}^{(2)}$
return $\hat{m}$

Figure 3.4: Stage III of the SSV2 algorithm.

Stage I of the SSV2 method covers all estimated IDs $\hat{m} < s^{(1)}_{\max}$. In the first FOR-loop, the nearest neighbor sets $\mathcal{S}_i$, the bounding ball radii r_i and the average simplex volumes $\overline{V}_i(s)$ for $s = 1, \ldots s^{(1)}_{\max}$ are computed, all for fixed $k = k_1$. The second FOR-loop examines each test dimension $d = 1, \ldots, s^{(1)}_{\max}$. Here, for each $s = 1, \ldots, d$, the quantities (3.116) are evaluated and in case the decision criterion (3.117) is fulfilled for any $s \geq 2$ (or for any s in case $d \leq 2$), the output value is fixed as $\hat{m} = \max\{d - 1, 1\}$.

If no intrinsic dimension $\hat{m} < s^{(1)}_{\max}$ has been found yet, the algorithm initiates stage II (fig. 3.3). In the first FOR-loop, the average simplex volumes $\overline{V}^{(z)}_i(s)$ are now computed for both NN values k_z, $(z = 1, 2)$, and for all $s = s^{(1)}_{\max}, \ldots s^{(2)}_{\max}$. The second FOR-loop examines each test dimension $d = s^{(1)}_{\max}, \ldots, s^{(2)}_{\max}$. The same quantities and decision criterion are utilized as above, this time for each $s = 2, \ldots, d$.

If this does not lead to some estimate $\hat{m} < s^{(2)}_{\max}$, stage III (fig. 3.4) is initiated. Here,

for each $z = 1, 2$ and for test dimensions $d = s^{(2)}_{\max}+1, \ldots, d_{\max}$, our test quantities (3.116) are computed, and the decision criterion (3.117) now potentially yields two (different) estimates $\hat{m}^{(z)}$, $z = 1, 2$. In the final step, the two estimates $\hat{m}^{(1)}$ and $\hat{m}^{(2)}$ are combined as follows. If both estimates have been assigned a value less than D, the final estimate is defined as

$$\hat{m} = \begin{cases} \hat{m}^{(1)} & \text{if } \hat{m}^{(1)} \leq k_1, \\ \frac{1}{2} \cdot \left(\hat{m}^{(1)} + \hat{m}^{(2)} \right) & \text{if } \hat{m}^{(1)} > k_1 \ \wedge \ \hat{m}^{(2)} \leq k_2, \\ \hat{m}^{(2)} & \text{otherwise.} \end{cases} \tag{3.118}$$

In case that only $\hat{m}^{(2)} < D$, we let $\hat{m} = \hat{m}^{(2)}$. The case where only $\hat{m}^{(1)} < D$ does not occur in practice.

Note that the estimation process of stage III is generally based on theoretical considerations (see subsection 3.4.3), while the shortcomings of using low-dimensional simplex volumes to estimate high intrinsic dimensionalities have been highlighted in 3.4.4. Consequently, if the estimated ID $\hat{m}$ is considerably larger than the maximum simplex volume $s^{(2)}_{\max}$, we do not expect the SSV2 results to be perfectly on par with the results produced by the SSV1 approach.

Another remark on both SSV methods should be given concerning the choice of the parameters C, the constant number of test simplices, and ϵ, the tolerance parameter.

To improve performance, the number of random test simplices could be chosen adapted to the current combination of variables. For example, when sampling triangles from $\mathcal{S}_i$ with size $|\mathcal{S}_i| = k_{\min} = 12$, there are only $\binom{12}{3} = 220$ possible different choices, hence less than $C = 1000$. However, the speed-up due to those savings is negligible in practice, as our tests have shown. For reasons of clarity, we therefore opted for a fixed number C.

When it comes to the tolerance parameter $\epsilon > 0$, our choice of a constant value $\epsilon = 0.05$ is surely suboptimal in some scenarios. It is reasonable to presume that there are theoretical considerations that the (in some sense optimal) value of ϵ should depend on the ambient dimension D and on the current test dimension d. In practice, we experience the following trend. For "uncomplicated" datasets (i.e., intrinsically low-dimensional, low noise level and curvature), ϵ can be selected from a rather large interval $[0, 0.5]$ to reliably produce correct estimation results. For more challenging datasets (higher intrinsic ID and noise level), the fine tuning of the proper selection of ϵ becomes a cumbersome task and is probably impossible without further knowledge of noise levels, manifold curvature, or sampling density. Eventually, to keep our results more easily reproducible, we settled for a constant ϵ.

3.4.6 Complexity analysis of the SSV methods

The complexity analysis of both SSV methods involves the investigation of three non-trivial steps. These are the determination of each nearest neighbor set $\mathcal{S}_i$, the computation of the corresponding minimum bounding ball $B(\mathcal{S}_i)$, and the evaluation of a simplex volume $\mathrm{vol}\,(\Delta_t)$.

The construction of the *kd*-tree $\mathbb{T}_{\mathcal{X}}$ can be done in $\mathcal{O}(N \cdot (D + \log N))$. The computation of the local subsets $\mathcal{S}_i$ is in fact equivalent to the so-called k-all-nearest-neighbors problem and could thus be realized using specialized approaches e.g. as suggested in [Vai89] or [SSV07]. However, we opted for a more basic approach. In our case, note that for a balanced tree, finding a single nearest neighbor is an $\mathcal{O}(D \log N)$ operation. Let k be the number of NNs to be found, this adds up to a total complexity of $\mathcal{O}(k \cdot D \cdot N \log N)$.

The computation of the corresponding minimum bounding ball $B(\mathcal{S}_i)$ is performed using the algorithm presented in [LK13]. The associated complexity is $\mathcal{O}(k \cdot D/\delta + D/\delta^3)$, where k is the respective number of points in $\mathcal{S}_i$ and $\delta = 0.001$ is the approximation parameter. For all points $\boldsymbol{x}_{i=1,\dots,N}$, this results in $\mathcal{O}(N \cdot (k \cdot D/\delta + D/\delta^3))$.

Finally, as shown in section 3.3, the evaluation of a total number of C volumes of s-dimensional simplices, with their vertex points drawn from the local set $\mathcal{S}_i$, can be accomplished in roughly $\mathcal{O}(D \cdot k^2 + C \cdot s^3)$ time complexity.

For the SSV1 method, the (practical) time complexity heavily depends on the estimated intrinsic dimension $\hat{m} \approx m$ of the dataset. The algorithm increments the test variable d in each step until $(d-1)$ is selected as the estimation result. For a fixed d, the time complexity is thus given by

$$\mathcal{O}(k_d \cdot D \cdot N \log N) + \mathcal{O}(N \cdot (k_d \cdot D/\delta + D/\delta^3)) + \mathcal{O}(N \cdot (D \cdot k_d^2 + C \cdot d^3)), \quad (3.119)$$

where $k_d = \max\{d + 3, k_{\min}\}$ is the current number of NNs and C is the fixed number of local test simplices, with default values $k_{\min} = 12$ and $C = 1000$, respectively. Hence, the overall complexity of the SSV1 method sums up to

$$\sum_{d=1}^{\hat{m}} \left[\mathcal{O}\left(N \cdot D \cdot (k_d \cdot \log N + k_d/\delta + 1/\delta^3 + k_d^2) + N \cdot C \cdot d^3 \right) \right]. \quad (3.120)$$

On the positive side, one can observe that the complexity is just linear in the ambient dimension D. However, for growing values of D, the nearest-neighbor search based on the *kd*-tree becomes more and more inefficient and often dominates the whole runtime complexity. Nevertheless, this is an issue of all IDE methods based on the analysis of nearest neighbors. The effective computation time of the approximate minimum bounding ball also grows with D, still, its influence on the overall runtime is usually negligible compared to the NN search. Finally, the complexity of the simplex volume computation is quickly dominated by the factor d^3 for higher IDs $m > 10$, resulting in runtimes notably higher than those of methods relying on distance computations only.

The SSV2 method attenuates this drawback for high IDs to some extent. Clearly, its time complexity is dominated by stage I and stage II. The same considerations as for the SSV1 approach lead to an overall complexity of the SSV2 method of

$$\sum_{d=1}^{s_{\max}^{(2)}} \left[\mathcal{O}\left(N \cdot D \cdot (k_{\max} \cdot \log N + k_{\max}/\delta + 1/\delta^3 + k_{\max}^2) + N \cdot C \cdot d^3 \right) \right], \quad (3.121)$$

where $s_{\max}^{(2)} = 10$, $k_{\max} = \max\{k_1, k_2\} = k_2 = 30$ and $C = 1000$ in our setting. For $\hat{m} < s_{\max}^{(1)}$, the sum in (3.121) of course is capped at $d = s_{\max}^{(1)}$. In practice, the user can select suitable values $s_{\max}^{(1)}$ and $s_{\max}^{(2)}$ to gauge precision versus speed.

Now, after we introduced the theory as well as the implementation details of our SSV algorithms, the upcoming section is dedicated to numerical experiments and comparisons with other IDE methods.

3.5 Numerical Results

In this section, we present the numerical results of various recent approaches for intrinsic dimension estimation (IDE), where our considerations will cover multiple different topics. The main focus of most IDE methods are datasets of relatively low ID values $m < 10$, but with a rather high ambient dimension D. Recently however, more and more practical scenarios have emerged where the data are also of higher intrinsic dimensions $m > 10$. Our experiments will cover multiple low- and high-dimensional cases for both synthetic as well as real-world datasets. Before we come to the experimental results, let us first consider several aspects making ID estimation more challenging, including noise in the data, curvature of the underlying manifold, and some (partly counterintuitive) phenomenons of data in high-dimensional spaces.

3.5.1 Challenges of dimension estimation

Data originating from real-world measurements is typically tainted with noise. This noise is usually modeled via some Gaussian distribution function, and in this case, it is referred to as *Gaussian noise*.

The so-called "swiss role" dataset has already been introduced in subsection 2.2.2 in the context of the ISOMAP dimensionality reduction algorithm. The left column of figure 3.5 shows three instances of the swiss role, sampled with 2000 points each, tainted with Gaussian noise of increasing standard deviation $\sigma = 0.01, 0.03, 0.05$. The first plot with Gaussian noise of standard deviation $\sigma = 0.01$ still clearly reflects an underlying two-dimensional object, while from the visual appearance, it is hard to tell whether the third variant with $\sigma = 0.05$ is intrinsically two- or three-dimensional. Furthermore, noise usually affects all D dimensions of the ambient space making the estimation of the intrinsic dimension even harder for high values of D.

A second aspect that must be taken into account when generating test data is the number of sampling points. Both, high curvature of the underlying manifold and a high intrinsic dimension require a large number of sampling points. To illustrate this effect, let us consider a k-times twisted Möbius strip in $\mathbb{R}^3$, given by the following mapping

$f_{\text{mo}} : \mathbb{R}^2 \to \mathbb{R}^3$,

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \mapsto \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} \left(1 + \frac{y_1}{2} \cdot \cos\left(k \cdot \frac{y_2}{2}\right)\right) \cdot \cos(y_2) \\ \left(1 + \frac{y_1}{2} \cdot \cos\left(k \cdot \frac{y_2}{2}\right)\right) \cdot \sin(y_2) \\ \frac{y_1}{2} \cdot \sin\left(k \cdot \frac{y_2}{2}\right) \end{bmatrix}, \tag{3.122}$$

where $y_1 \in [-1, 1]$, $y_2 \in [0, 2\pi)$ are uniformly distributed.

The second column of figure 3.5 features three instances of the 10-times twisted Möbius strip, each with a different sampling number. While the two-dimensional structure of the Möbius strip is clearly visible for a sample size of 10000 points and still well recognizable for 1000 points, the sampling with only 100 points leaves just 10 points per "curl" making a purely visual identification of the underlying structure virtually impossible.

Another problem is the so-called *edge effect*, which has been described e.g. in [VD95], [CV02], and analyzed in [BGRS99]. As already explained in subsection 2.1.1, data sampled from within an m-dimensional ball will tend to accumulate close to its boundary, i.e., the associated $(m-1)$-sphere, for $m \to \infty$. The number of vertices of an m-dimensional cube grows exponentially with m, as a consequence, the sampled points concentrate in the corners. Basically, all datasets of high *intrinsic* dimensionality m are affected by the edge effect. This leads to an underestimation of the true dimensionality, since the boundary of a manifold is of dimension $m-1$. The authors of [CRH10] try to face this problem by introducing different weights for local dimension estimates before combining them into a global estimate. For this purpose, they assign a measure of "depth" to each data point, where points with higher depth are supposed to be farther away from the boundary / edge. Subsequently, local estimates associated with points of higher depth are given higher weights. The authors verify that this procedure indeed exhibits positive effects on estimation results for a rather low intrinsic dimension $m = 6$. However, they also show that for increasing ID m, the effect vanishes since the minimum and maximum depth of all data points converges to the same value.

Finally and naturally, the curvature of the underlying manifold plays an important role when it comes to the reliability of ID estimators. In [HA05], the authors consider the extreme example of a highly oscillating sinusoid defined by $f_{\text{si}} : [0, 2\pi) \to \mathbb{R}^3$,

$$y \mapsto \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} \sin(y) \\ \cos(y) \\ \sin(150y)/10 \end{bmatrix} \tag{3.123}$$

to demonstrate this effect. The above sinusoid can be best described as a "ring" in $\mathbb{R}^3$ with small height of 0.2, where the third coordinate oscillates 150 times around 0. When comparing a plot of the sinusoid sampled with 600 points with a plot of a circle with uniform noise of 0.1 in the third coordinate, again sampled with 600 points, no evident difference can be seen (compare figure 3.6). The authors of [HA05] highlight that, nevertheless, their method is able to estimate the correct ID values of 1 and 2,

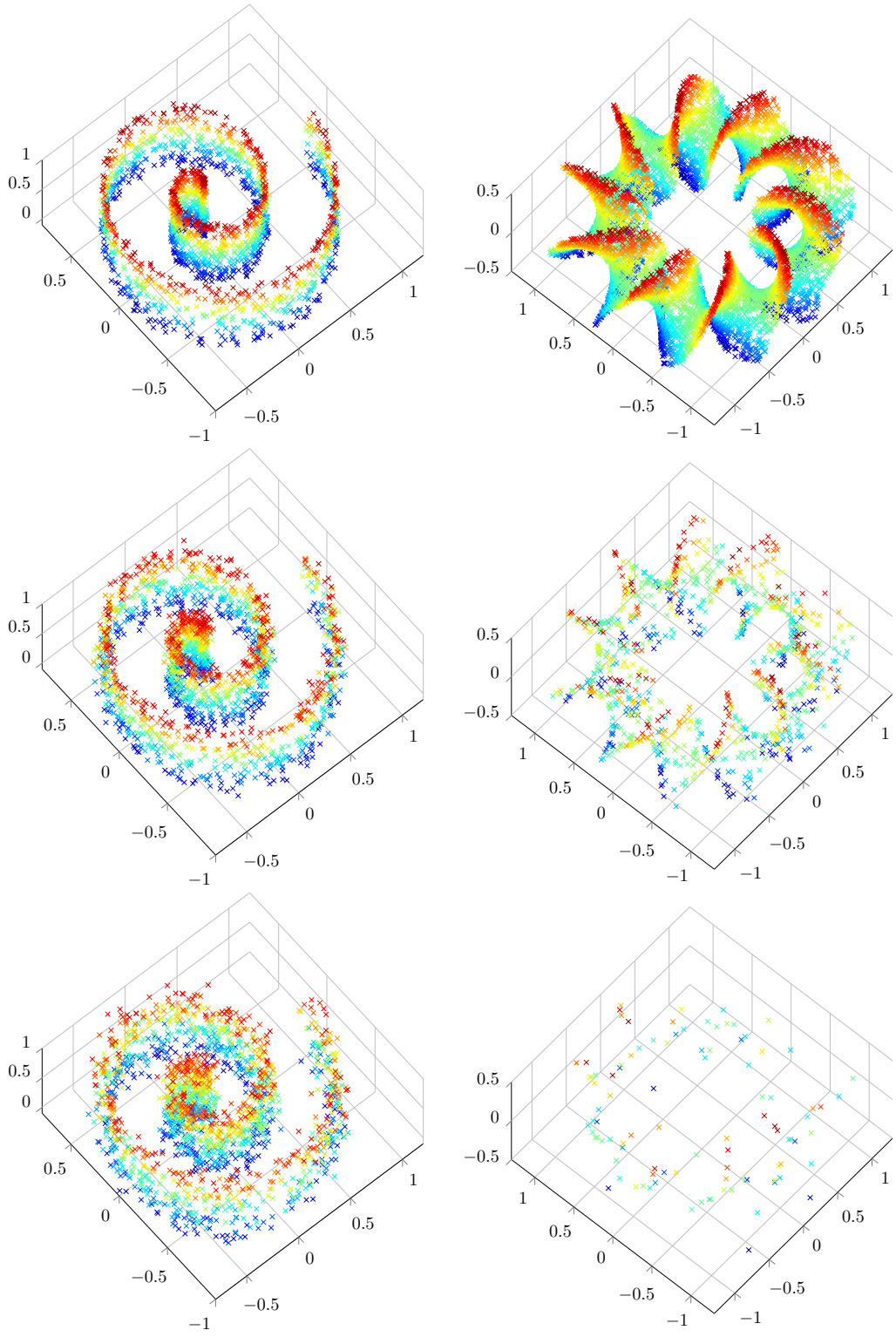

Figure 3.5: *Left:* Swissrole with Gaussian noise of standard dev. $\sigma = 0.01, 0.03, 0.05$. *Right:* 10-fold Möbius strip sampled with 10000, 1000 and 100 points.

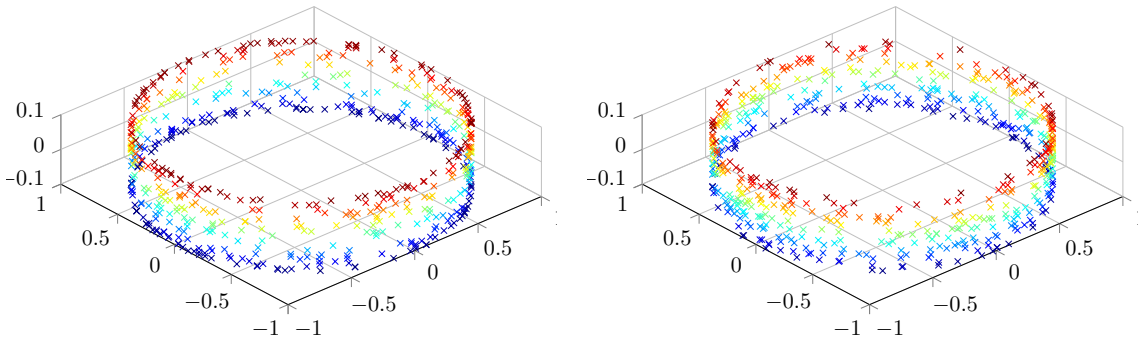

Figure 3.6: *Left:* Highly oscillating sinusoid sampled with 600 points. *Right:* Circle with uniform noise of 0.1 in the third coordinate, sampled with 600 points.

respectively, in the majority of cases. Our point of view for this scenario is slightly different, though. With a sampling number as low as 600 points, which, on average, leaves 4 points per oscillation, the dataset rather corresponds to a space-filling curve, where the space in this case is the two-dimensional "strap". Thus, an estimate of $\hat{m} = 2$ for this special case might not be as wrong as suggested by the authors.

The crucial conclusion that should be drawn from this example is that highly curved manifolds require a large number of sample points.

3.5.2 Configuration of the tested methods

We now present the detailed parameter configuration of our testbed. Table 3.8 shows all tested methods, introduced in subsections 3.2.2 and 3.4.5, with their corresponding original publication and the chosen parameter settings. All parameter values where selected to yield the best possible estimation results in each case. For the GCD, MLE and ANC methods, the associated parameters have been fixed according to the suggestions in the respective publications. This seems reasonable since our own tests of those methods with different parameter values showed no substantial variation in the results. For the CD and the DD method, we found that values larger than $L = 20$ or $k = 30$, respectively, did only lead to increasing runtimes, while the estimates did not become more precise.

The ESS method comes with no recommendation for a practical selection of its parameter k. This is why we conducted multiple tests with different values of k ranging between 15 and 50, which led us to the conclusion that the ESS estimates tend to be slightly better for higher values of k. Since all the experiments in the original publication [JSF15] are performed for local datasets of size $k = 50$, we fixed the parameter accordingly.

The minimum number of NN points for the SSV1 method proposed in this thesis is fixed as $k_{\min} = 12$; further, the tolerance parameter is set to $\epsilon = 0.05$, which is relatively small. As already explained above, this allows for the precise estimation of high intrinsic

abbr.	method	citation	parameters
CD	correlation dimension	[GP83]	no. of intervals $L = 20$
GCD	generalized correlation dimension	[HA05]	no. of sub-sample sizes $N_{\text{sub}} = 5$
MLE	maximum likelihood estimation	[BL05]	min./max. no. of NNs $k_1 = 10$, $k_2 = 20$
DD	distribution of distances	[CNBYM01]	no. of NNs $k = 30$
ANC	angle and norm concentration	[CCB$^+$14]	no. of NNs $k = 10$
ESS	expected simplex skewness	[JSF15]	no. of NNs $k = 50$
SSV1	sample simplex volumes (1)		min. no. of NNs $k_{\min} = 12$, tol. $\epsilon = 0.05$
SSV2	sample simplex volumes (2)		$s^{(1)}_{\max} = 5$, $s^{(2)}_{\max} = 10$, $k_1 = 12$, $k_2 = 30$

Table 3.8: IDE methods with their parameter configurations

dimensions, however the ID of data tainted with noise is likely to be overestimated in certain cases. Besides, the highest possible estimated ID value is limited to 63, due to the growing costs for the computation of the high-dimensional simplex volumes.

Concerning our SSV2 method, the maximal dimension of test simplices is chosen as $s^{(1)}_{\max} = 5$ for stage I and $s^{(2)}_{\max} = 10$ for stage II. While smaller values would reduce the time complexity of the estimation process, larger values could yield better estimates for datasets of higher intrinsic dimensionality. As for the SSV1 method, the tolerance parameter is fixed as $\epsilon = 0.05$. The two different numbers of nearest neighbors are chosen as $k_1 = 12$ and $k_2 = 30$.

When it comes to the actual implementations, we rely on the `C++` code of the GCD method provided online at [HA16] and on the `Matlab` code of the ANC method provided at [Lom13]. All other methods have been implemented in `C++` carefully following the descriptions presented in subsections 3.2.2 and 3.4.5, respectively.

Finally, for the sake of accuracy, we remark that the precise implementations of the methods DD, ESS, MLE, SSV1, and SSV2 have been slightly modified in the following way, with the sole purpose of accelerating our computations. Note that all those methods calculate certain local quantities at *each* point of the dataset, which are averaged subsequently; the final result only depends on the respective average in each case. Thus, for a sufficiently large number of data points N, it seems natural to assume that the evaluation of the quantity at only a reduced, but well selected number of points will yield the same final output than the inclusion of all points. Accordingly, each of the five methods features a *kd*-tree (as described in the beginning of subsection 3.4.5) which accelerates the computations of nearest neighbors in the first place. For all datasets with $N \leq 1000$, the local quantities are computed at all points, as expected. For larger datasets, a certain number of test points is chosen from each tree leaf. More precisely, if the current leaf size is n_l, the number of randomly selected test points equals $n_t = \max\{10, \lceil n_l/8 \rceil\}$. Note here that, in the tree construction process, a node is split only if its size is larger than 100; consequently, leaf sizes vary roughly between 50 and 100 points. Eventually, for each tree leaf, the local quantities are only evaluated for the corresponding test points,

instead of for all points. After thorough testing, we found that this process yields the exact same estimation results for each of the five methods in every test scenario, while tremendously reducing the overall runtimes. Let us yet emphasize that a completely random sampling of test points from the whole dataset might not work as well, while our selection procedure of test points from each leaf ensures that no local region is missed out.

3.5.3 Technical prerequisites

In order to get reliable empirical results, we proceed in the following way (this is valid for all upcoming synthetic datasets, unless noted otherwise). We choose a constant number of sampling points $N = 10000$, which is sufficiently high for all our tests and also a test size that all methods can still handle in a reasonable amount of time. In certain cases, when relevant for a deeper understanding, tests with lower sample sizes will be performed and discussed.

The test points are sampled by means of the popular pseudo-random number generator known as "Mersenne Twister", see [MN98]. The corresponding pseudo-random numbers are equidistributed in 623 dimensions and have a period of $\approx 4 \cdot 10^{6001}$; we utilize a variant named "MT19937" from the GNU Scientific Library [GNU17]. This library also includes the assistant routines `gsl_rng_uniform`, `gsl_ran_gaussian`, and `gsl_ran_exponential` for getting random samples distributed according to the uniform, the Gaussian, and the exponential distribution, respectively.

Each of our experiments is repeated 10 times, meaning that 10 different random datasets are generated, which are then given as input to each method. Our presented results are the respective average outcomes.

Finally, note that some methods produce a floating point output for the ID estimate, while others come with an integer output value. In order to get a fair comparison, we perform a rounding for all floating point outputs to the nearest integer. The decimals in our results are due to the averaging of 10 experiments in each case.

3.5.4 Results from synthetic data

Symmetrical datasets: unit ball, cube, simplex, and isotropic Gaussian

We start with the analysis of the results for important basic datasets: we consider the three most elementary geometric objects in different dimensions:

- the unit m-ball: $B^m = B_1^m(\mathbf{0}) = \{\boldsymbol{x} \in \mathbb{R}^m : \|\boldsymbol{x}\| \leq 1\}$;
- the unit m-cube: $C^m = [0,1]^m \subset \mathbb{R}^m$;
- the unit m-simplex: $S^m = \{(x_0, \ldots, x_m) \in \mathbb{R}^{m+1} : \sum_{i=0}^m x_i = 1 \text{ and } \forall i : x_i \geq 0\}$.

The unit (or standard) m-simplex is part of an affine hyperplane of $\mathbb{R}^{m+1}$, its $m+1$ vertex points are given by $(1,0,0,\ldots,0),(0,1,0,\ldots,0),\ldots,(0,\ldots,0,1)$. Naturally, a rotation allows the representation of S^m in $\mathbb{R}^m$, and the only reason for defining the m-dimensional unit simplex in $\mathbb{R}^{m+1}$ is simplicity.

We consider datasets sampled uniformly from the interior of these m-dimensional objects as well as sampled uniformly on their particular $(m-1)$-dimensional boundary (or surface). The associated surfaces are the $(m-1)$-sphere, the union of $2\cdot m$ boundary $(m-1)$-cubes, and the union of $m+1$ boundary $(m-1)$-simplices, respectively.

The uniform sampling of the interior or the surface of the cube is a trivial task. Considering the ball and the sphere, we exploit the well-known fact that the multivariate Gaussian distribution $\mathcal{N}_m(\mathbf{0},\mathbf{1})$ is radially symmetric. Thus, given a random variable $\mathbf{Y}_m \sim \mathcal{N}_m(\mathbf{0},\mathbf{1})$, the variable $\mathbf{S}_m = \mathbf{Y}_m/\|\mathbf{Y}_m\|$ has the uniform distribution on the unit $(m-1)$-sphere (see e.g. [Mul59]). Furthermore, it is easy to see that $U^{1/m}\cdot\mathbf{S}_m$, where U has the uniform distribution on the interval $(0,1)$, is uniformly distributed on the unit m-ball.

Regarding the simplex, we refer to the following efficient and simple sampling method revisited by Rubinstein in [Rub82]. Consider the exponential distribution $\exp(\lambda)$ with rate parameter $\lambda > 0$ and its associated density

$$f(x) = \begin{cases} \lambda e^{-\lambda x} & \text{if } x > 0, \\ 0 & \text{otherwise.} \end{cases} \tag{3.124}$$

Now let $Y_0,\ldots,Y_m$ be independent $\exp(1)$ distributed random variables and define $Z = \sum_{i=0}^{m} Y_i$. Then, $\mathbf{X} = (X_0,\ldots,X_m) = (Y_0/Z,\ldots,Y_m/Z)$ is uniformly distributed on the unit m-simplex S^m. Note that — in contrast to our notion — in [Rub82], the m-simplex is in fact referred to as the "surface of the $(m+1)$-dimensional simplex". Uniform sampling of the boundary of the m-simplex is easily implemented via the above procedure, where in each sampling step, one of the $m+1$ coordinates is randomly selected to be zero.

The last dataset included in our first comparison is a point sample of the isotropic multivariate (m-dimensional) Gaussian distribution $\mathcal{N}_m(\mathbf{0},\mathbf{1})$ with variance matrix $\Sigma^2 = \sigma\cdot\mathbf{1} = \mathbf{1}$ as $\sigma = 1$. It is not surprising, but still noteworthy that all methods yield the same estimation results for different values of σ. This is due to the fact that for the isotropic Gaussian, all distances, and consequently all angles as well as volumes, are scaled proportionally with varying σ.

Now after sampling a particular set of points from the underlying m-dimensional structure, we embed it into D-dimensional space with a sufficiently high ambient dimension D and perform a random rotation. Without the embedding, overestimation of the intrinsic dimension would not be possible for most methods, since their estimation results are limited to the outer dimension D. To be exact, D was fixed as $D = 20$ for $m = 4, 10$, $D = 30$ for $m = 16$, and $D = m + 20$ for $m = 20, 30, 40$.

Figure 3.7 shows three-dimensional plots of selected datasets, i.e., the surface of the three-dimensional ball, cube, simplex, as well as the isotropic Gaussian. Here, the sam-

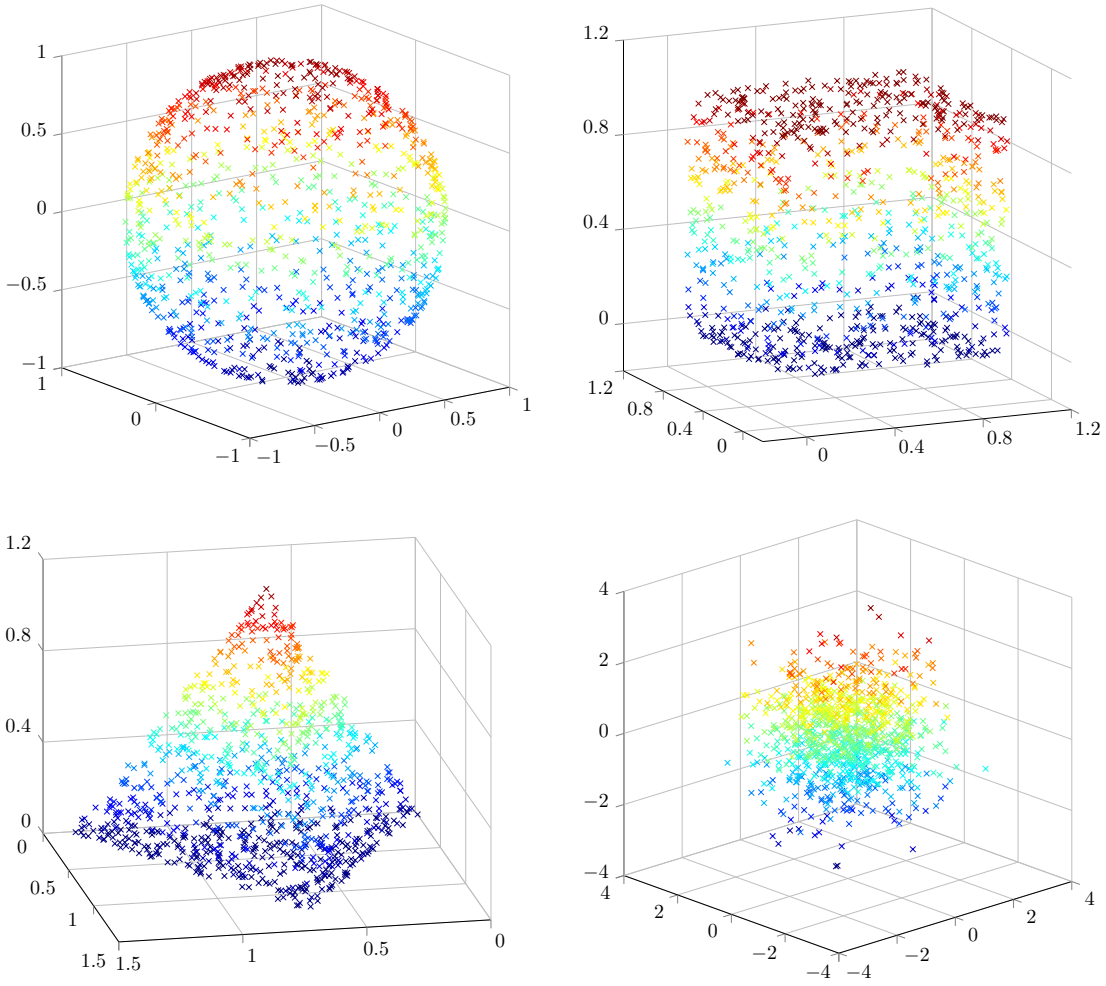

Figure 3.7: Symmetrical test datasets sampled with 1000 points each: three-dimensional unit ball, cube, simplex, and isotropic Gaussian. Here, for each of the three geometric objects, uniformly distributed points have been sampled on their two-dimensional surface. The unit simplex has been rotated and shifted in order to fit in $\mathbb{R}^3$.

pling size has been reduced to 1000 points for improved visualization.

Table 3.9 summarizes the experimental results and already allows first insights into the strengths and drawbacks of the different approaches on the one hand, and into some pitfalls of estimating higher IDs on the other hand.

To begin with, we observe that two groups of methods perform quite similarly so far. The first group (CD, GCD, MLE) achieves accurate results for low intrinsic dimensions $m \leq 10$, however for growing values $10 < m \leq 40$, the methods suffer from an increasing underestimation of the ID. The second group (ANC, ESS, SSV1) yields accurate results with only minor errors across the entire spectrum of m, for most datasets. In contrast, the DD approach generally produces overestimations which tend to increase for growing ID. Correspondingly, the estimation errors of the DD method averaged over all IDs m for a fixed dataset are larger than for any other method in the comparison; the sole exception is the m-simplex, where the tendency to overestimation seems to cancel out with a peculiarity of the dataset leading to underestimation of most other methods. The SSV2 method performs very similar to the SSV1 method for ID values $m \leq 16$; however, for larger intrinsic dimensions, minor tendencies of the SSV1 method to overestimation (ball, sphere) and underestimation (simplex) are amplified leading to less reliable results of the SSV2 approach.

A closer look at the outcomes for $m = 10$ and larger intrinsic dimensions for the different structures reveals some of the characteristics of IDE methods presented in subsection 3.5.1. First, the m-ball naturally represents the best-case scenario for nearly all approaches, due to similar model assumptions, which can be seen from the near perfect results of ANC, ESS and SSV1. The m-sphere is not a highly curved manifold, however its constant curvature in $(m+1)$ dimensions entails slight overestimations (ANC and SSV1) or considerable overestimations (DD and SSV2) for values of $m \geq 20$.

On the other hand, the m-cube with its 2^m vertices concentrates many sample points close to its boundary ("edge effect") resulting in a minor underestimation for the second group (ANC, ESS, SSV1) as well as a larger underestimation for the first group (CD, GCD, MLE). The surface of the hypercube is again perfectly estimated by the ESS and the SSV1 method.

The m-simplex is obviously the most challenging of the three geometric objects, due to its acute angles at the vertex points making it more spiky than the m-cube. Even the generally well-performing methods ANC and ESS suffer from a growing underestimation for increasing ID values of $m \geq 16$. Here, the SSV1 method clearly outperforms all its competitors for both datasets (the interior as well as the surface of the simplex), even though its estimation results are not completely perfect.

Finally, the isotropic high-dimensional Gaussian is known to have most of its weight concentrated in its tails (see e.g. [LV07]). We therefore expect some mild underestimation issues, which are encountered in our SSV1 and SSV2 methods. While the ESS algorithm yields correct estimates for all ID values m, the results of the ANC rather show a moderate overestimation. The remaining methods feature a similar behavior as discussed before.

Table 3.9: IDE results for symmetrical datasets: unit ball, sphere, cube (and its surface), simplex (and its surface), and isotropic Gaussian of varying dimension m

dataset	m	CD	GCD	MLE	DD	ANC	ESS	SSV1	SSV2
m-ball	4	4	4	4	5	4	4	4	4
	10	9.1	9	9	14	10	10	10	10
	16	13.8	14	13	23	16	16	16	16
	20	17.4	17	16	29	20	20	20	21
	30	23.2	23.6	22	44	30	29.9	30	32
	40	29.4	29.7	27	59	40	39.8	40	45.4
m-sphere	4	4	4	4	5	4	4	4	4
	10	9.4	9.9	9	15	10.1	10	10	10.4
	16	14.2	14.7	14	24.6	16.8	16	17	18
	20	17.2	17.6	16.9	31	20.5	20	21	23
	30	24.2	24.3	22.7	46	30.4	30.2	31	35
	40	29.8	30.2	28	61	40.5	40.1	41	50.8
m-cube	4	4	4	4	5	4	4	4	4
	10	9	9	8.6	13	9.9	9	9	9
	16	13.1	13.1	13	21	15	15	15	15
	20	15.7	16	15	27	19.1	19	19	19
	30	22.1	22	21	41	29.8	29	29	30.6
	40	27.7	27.2	26	55	40	39.1	39	35.8
surface of $(m+1)$-cube	4	4	4	4	5	4	4	4	4
	10	9.2	9.3	9	14	10	10	10	10
	16	14	14	13	23	16	16	16	16
	20	16.5	17	16	29	20	20	20	20.3
	30	22.3	22.7	21.8	43	30.3	30	30	32.9
	40	27.7	28	26.9	57.1	41.2	40	40	37.6
m-simplex	4	4	4	4	5	4	4	4	4
	10	8.7	9	8	12	9	9	9	9
	16	12.4	12.2	11	18	14	13.2	15	13
	20	14.6	14.6	13	22	17	17	18	16
	30	19.2	19	18	31.9	24.2	25	28	23.4
	40	23.4	23	22	41.8	31.5	33	37	31.9
surface of $(m+1)$-simplex	4	4	4	4	5.3	4	4	4	4
	10	9	9	8.6	13	10	9.9	10	10
	16	13	13	12	19.2	14.6	14	16	14
	20	15.3	15.1	14	23.1	17.5	17.4	19	17
	30	19.7	19.8	18	33	25.1	25.4	28.6	24
	40	23.8	23.6	22	43.2	32	34	38	33.1
m-variate Gaussian	4	4	4	4	5	4	4	4	4
	10	9	9	10	12.7	11	10	9	9.9
	16	13.7	13.2	14	20	17.5	16	15	14
	20	16.3	15.9	17	25	21.7	20	19	17.7
	30	21.5	20.9	22	37.2	32.3	30	29	27.5
	40	26.8	25.3	27	49.7	42.9	40	38	39.6

Table 3.10: Average errors of IDE methods for symmetrical datasets in table 3.9

	CD	GCD	MLE	DD	ANC	ESS	SSV1	SSV2
abs. error	4.776	4.788	5.417	6.352	1.250	0.943	0.676	2.164
rel. error	0.175	0.173	0.201	0.314	0.049	0.039	0.031	0.081

Table 3.10 presents the average errors of the different estimation methods for our first testbed of symmetrical datasets. Let $T = 42$ be the number of all test datasets and let $\hat{m}_\Psi(t)$ denote the estimate of method Ψ for the tth dataset with ID $m(t)$. The absolute and relative errors are then defined as

$$\mathbf{E}_{\text{abs}}(\Psi) = \frac{1}{T}\sum_{t=1}^{T} |\hat{m}_\Psi(t) - m(t)|, \qquad \mathbf{E}_{\text{rel}}(\Psi) = \frac{1}{T}\sum_{t=1}^{T} \frac{|\hat{m}_\Psi(t) - m(t)|}{m(t)}. \tag{3.125}$$

We conclude that for our first experiment with standard symmetrical datasets, the three methods ANC, ESS and SSV1 perform significantly better than the remaining estimators. As can be seen from the average errors, our SSV1 approach is indeed the most reliable of all compared estimators. This is also the case for considering the error in the maximum norm, since we have $|\hat{m}_{\text{SSV1}}(t) - m(t)| \leq 2$ for all datasets except for a single one, which is the 40-dimensional simplex. Furthermore, our SSV2 method performs very well for lower ID values $m \leq 16$ and still considerably better on average than the CD, GCD, MLE and DD approaches.

Non-symmetrical datasets: ellipsoid, rectangle, paraboloid, and anisotropic Gaussian

The symmetrical datasets considered above are an important benchmark set, however, in practice, data will often rather be distributed in a non-symmetrical, anisotropic manner. In the field of data mining, some techniques propose a general pre-processing of the data to equalize either the spread or the variance in each coordinate direction, thus possibly reducing the level of anisotropy. However, there are two important arguments against this pre-processing. First of all, the anisotropy in the original dataset can in fact contain crucial information about the relationships between the different components, that should not be eliminated in the first place. In addition, the "distortion" of the data might not be parallel to the axis, not even linear, which is why the identification and removal of the unwanted distortion might be much more complex than the actual analysis task itself.

Nevertheless, high levels of anisotropy lead to a fundamental conflict for every estimator of the intrinsic dimension. Consider a rectangle with its two side lengths $l_1 = 1$ and $l_2 = 1000$. In theory, this rectangle is a two-dimensional object, while in practice,

it clearly depends on the number of sample points and the model assumptions about the noise level whether it would be considered a one- or two-dimensional dataset. For this case, an estimator could of course choose a real number $1 < \hat{m} < 2$ as output value to gauge the level of anisotropy. However, this strategy can not be generalized in a meaningful way for high-dimensional data.

In summary, solid estimators should be able to correctly classify datasets with rather small levels of anisotropy, while for higher levels, we expect a tendency towards underestimation of the true underlying dimension.

Our second set of test datasets consists of the following four non-symmetrical objects:

- the m-dimensional ellipsoid with semi-axes $a_1 = 1, a_2 = 2, \ldots, a_m = m$;
- the m-dimensional rectangle with side lengths $l_1 = 1, l_2 = 2, \ldots, l_m = m$;
- (a segment of) the m-dimensional paraboloid, defined by $y_{m+1} = y_1^2 + \cdots + y_m^2$;
- the m-dimensional anisotropic Gaussian with variance $\Sigma = \frac{1}{4}\operatorname{diag}(1, 2, \ldots, m)$.

Note first that for the ellipsoid and the rectangle, the level of anisotropy increases with growing ID m, since the ratio of the smallest semi-axis (or side) to the longest is $1 : m$. The anisotropic Gaussian features a similar characteristic due to its diagonal variance matrix Σ, where the corresponding 1-dimensional Gaussians are uncorrelated, but of growing variance $\sigma^2 = i/4$ for $i = 1, \ldots, m$. These three datasets can easily be sampled uniformly in an analogical way as their symmetric counterparts.

The data points of the m-dimensional elliptic paraboloid on the other hand are sampled as proposed in [BQY13]. Starting with *i.i.d.* exp(1) random variables $E_0, E_1, \ldots, E_m$, i.e., variables distributed according to the exponential distribution with rate parameter $\lambda = 1$, define the variables $Y_i = (1 + E_i/E_0)^{-1}$ for $i = 1, \ldots, m$. Finally, let $Y_{m+1} = Y_1^2 + \cdots + Y_m^2$. Note that the paraboloid has a non-constant curvature and the sampling method employed here yields a non-uniform distribution (compare fig. 3.8).

The numerical results from table 3.11 for the ellipsoid, the rectangle and the Gaussian indicate that the estimates of the SSV1 method deteriorate to a lesser extent than those of the ANC and ESS methods when it comes to anisotropic datasets compared to their isotropic (symmetric) counterparts. This underlines the superiority of our SSV1 method. Expectedly, the estimates of the CD, GCD and MLE methods become worse for higher values of m, a drawback that the SSV2 method shares to a lesser degree. The DD method on the other hand seems to benefit from a compensation of its general tendency to overestimation. The paraboloid dataset stands out probably due to its non-uniform sampling. Obviously, CD and GCD entirely fail to detect the true underlying manifold structure, while the DD approach again overestimates the intrinsic dimension as previously. The SSV1 approach clearly shows the best estimates for this example.

Finally, the average error as listed in table 3.12 confirms that our SSV1 method outperforms its closest competitors ANC and ESS, while the SSV2 method is slightly superior as compared to ESS. The DD algorithm's overestimation issue basically cancels out

Table 3.11: Numerical IDE results for non-symmetrical datasets: ellipsoid, rectangle, paraboloid, and anisotropic Gaussian of different dimensions

dataset	m	CD	GCD	MLE	DD	ANC	ESS	SSV1	SSV2
	4	4	4	4	5	4	4	4	4
	10	8.5	8.9	8	12	9	8	9	9
m-dim.	16	12	12	11	18	13.5	13	14	13
ellipsoid	20	14.3	14.4	13	22	16.2	15	17	16
	30	19	18.9	17	31	23	22	25	22
	40	23.5	23	21	39.6	30.1	28	32	30
	4	4	4	4	5	4	4	4	4
	10	8.1	8	8	11	9	8	9	9
m-dim.	16	11.5	11.8	11	17	13	12	14	12
rectangle	20	13.3	13.4	12	20	15	14.6	17	15
	30	17.7	17.8	16	29	21.7	21	24	21
	40	21.6	21.6	20	37.5	27.7	26.9	31.1	28.2
	4	2.4	2	4	5	4	4	4	4
	10	2.1	3	8	11.9	9	9	9	9
m-dim.	16	2.1	3.7	11	18	14	14	15	13
paraboloid	20	2.1	4	13	22.8	16.8	17	19	16.4
	30	2.1	4.9	17	33.7	23.8	25	28	25.1
	40	2.3	5.2	21	44.5	30.1	33	38	35.2
	4	4	4	4	5	4	4	4	4
	10	8.4	8	8	11	9.8	8	9	9
m-variate	16	11.7	11	12	16	14	12.2	13	12
anisotropic	20	13.9	13	13.1	19	16.9	15	16	14
Gaussian	30	17.8	17	18	27	23.3	21	24	20
	40	20.8	20	21	34.9	30.2	27.8	31	26.3

Table 3.12: Average errors of IDE methods for non-symmetrical datasets in table 3.11

	CD	GCD	MLE	DD	ANC	ESS	SSV1	SSV2
abs. error	9.700	9.433	7.704	1.746	4.079	4.729	2.913	4.533
rel. error	0.399	0.390	0.295	0.116	0.152	0.190	0.116	0.178

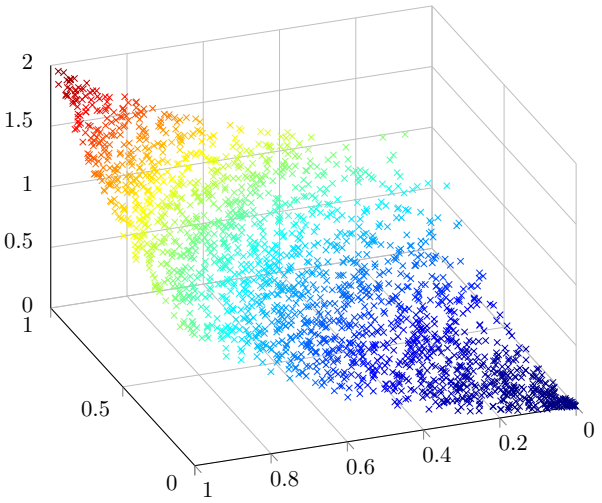

Figure 3.8: Two-dimensional paraboloid sampled with 2000 points.

with a general trend to underestimation for anisotropic examples; hence, its seemingly advantageous results must be put into perspective with its mediocre performance for the standard datasets in the first experiment. Therefore, when combining the results for symmetrical and non-symmetrical data, the SSV1 approach comes out as the most reliable IDE method so far.

Undersampling

As already discussed in subsection 3.5.1, two of the main issues that aggravate dimension estimation (or sometimes render it impossible) are undersampling and noise. First, we investigate the effects of undersampling by repeating our first two numerical experiments (symmetrical and non-symmetrical data), where the number of points per dataset is now fixed as $N = 100$.

The corresponding results for symmetrical datasets can be found in tables 3.13 and 3.14. Comparing the estimates to those of our first experiment with 10000 points, we

Table 3.13: Average errors of IDE methods for symmetrical datasets (undersampling) from table 3.14

	CD	GCD	MLE	DD	ANC	ESS	SSV1	SSV2
abs. error	9.471	7.074	9.400	8.569	1.555	0.505	0.769	3.240
rel. error	0.395	0.277	0.385	0.400	0.074	0.038	0.044	0.134

Table 3.14: IDE results for symmetrical datasets (sampled with 100 points each): unit ball, sphere, cube (and its surface), simplex (and its surface), and isotropic Gaussian of varying dimension m

dataset	m	CD	GCD	MLE	DD	ANC	ESS	SSV1	SSV2
m-ball	4	3.3	3.8	3.3	5	3.8	4	4	4
	10	6.8	8.6	7	14.6	10.1	10	10	10
	16	9.9	11.8	10.1	25.5	15.6	15.7	16	17.5
	20	12.1	14	11.8	33.1	19.4	20	20	23.6
	30	16.8	20.4	15.6	52.1	29.7	30	30	35.2
	40	19.2	25.3	18.5	70.2	40.3	39.4	40	52.6
m-sphere	4	3.7	4.3	4	6	4	5	4	4
	10	7.5	8.4	7.8	17.5	10.1	11	11	11.6
	16	10.7	14	10.6	29.2	17.3	16.9	17	20.2
	20	12.2	17.4	12.2	36.8	20.9	20.9	21	27.1
	30	17.1	21.6	16.1	56.6	30.7	31.1	31	41
	40	19.5	26.5	19.1	75.7	40.1	40.9	41	59.6
m-cube	4	3	3.8	3.1	4.9	3.4	4	3.8	3.6
	10	7	8	7.1	13.7	10.1	10	9.9	9.9
	16	10.2	12.4	10	23	16	15.9	15.7	15.9
	20	11.8	14	11.8	29.4	20.3	20	19.3	20.1
	30	15.3	17.8	15	45.4	30	30	29	33.7
	40	18.2	22.2	18	60	40.8	40.5	39	36.9
surface of $(m+1)$-cube	4	3.7	3.9	3.9	6	3.9	5	4.3	4.3
	10	7.2	8.9	7.6	16.2	10.7	11	11	11
	16	10	12.5	10.3	25.7	17.6	16.9	16.7	17.7
	20	12.5	14.7	12.2	32.4	21.9	20.9	20.5	23
	30	14.9	18.4	15.4	47	30.1	30.6	30.1	35.3
	40	19.6	22.9	18.2	63.1	41.5	41.1	40	37.7
m-simplex	4	3	3.7	3	4.1	3.1	4	3.1	3.1
	10	6.3	7.6	5.9	10.8	8.5	9.3	9	8.8
	16	8.7	10.3	8.3	17.3	13	15	14.7	12.1
	20	9.7	11.5	9.9	21.1	16.5	19.1	18.1	15.9
	30	11.2	15.5	12.3	31.6	22.3	28.4	27.6	23.5
	40	14.1	16.9	15	43.2	29.4	38.5	37.2	33.5
surface of $(m+1)$-simplex	4	3.6	4.2	3.5	5.5	3.9	4.7	4	4
	10	6.4	7.7	6.5	12.3	9.1	10.2	10	9.9
	16	9	10.7	8.9	18.7	14.5	16.1	15.5	13.6
	20	9.8	12.9	10.2	23	17.3	19.9	19.4	16.7
	30	12.2	14.5	12.7	32.4	23.6	29.5	28.6	24.5
	40	15.3	18.2	15	44.1	29.6	39.5	38.1	36.7
m-variate Gaussian	4	3.6	3.7	4	4.4	4	4	3.8	4
	10	6.9	7.9	7.5	11.8	11.1	10	9.3	9.8
	16	9.7	11.1	9.9	18.7	16.4	15.9	15	13.6
	20	11.3	12.9	11.8	24.8	19.4	19.9	19	17.2
	30	13.2	17.4	14.6	36.4	29.5	29.9	28.2	28.2
	40	16	21.6	17.5	50.6	39.2	40.3	38	36.3

Table 3.15: IDE results for non-symmetrical datasets (sampled with 100 points each): ellipsoid, rectangle, paraboloid, and anisotropic Gaussian

dataset	m	CD	GCD	MLE	DD	ANC	ESS	SSV1	SSV2
	4	3.4	3.6	3	4	3.2	3	3.1	3.4
	10	5.7	6.5	6	9.8	7.9	7	8	7.9
m-dim.	16	8.3	10.7	8	15.5	12	10.4	12.9	11.5
ellipsoid	20	9.6	11.3	8.9	18.8	13.9	13	15.8	13.6
	30	12.3	14.6	11.3	28	19.2	18.2	22.3	20
	40	14	17	13.5	35.8	26.9	24.4	29.6	27.8
	4	3	3.4	3	4	3	3	3	3
	10	5.3	6.6	5.4	9.3	8	6.9	8	8
m-dim.	16	7.1	9.5	7.6	14.3	11.5	10	12.1	11.7
rectangle	20	8.7	11	8.6	18.1	13.6	12.3	15	13.2
	30	11.5	13.9	10.8	26.8	19.8	17.8	22	20.2
	40	11.7	16.5	12.9	35.2	24.9	23.2	28.9	26.9
	4	2.4	2.7	3	3.7	3.1	3	3.1	3.1
	10	3.8	4.6	4.4	5.3	6.7	2.9	8.6	6.7
m-dim.	16	1.3	5.4	4.9	5.2	8.7	2.4	12.8	9.1
paraboloid	20	1.4	5.9	5	4.5	9	2	14.6	10.5
	30	1	6.4	4.8	3.4	9.6	2	15.1	9.5
	40	1	7.3	4.3	3.1	9.9	2	15.6	8.5
	4	3.4	3.7	3.6	3.9	3.9	3.2	3.1	3
	10	5.5	7	6	8.9	7.9	7	8	7.2
m-variate	16	7	9.3	7.8	13	12.1	10.6	12	11.5
anisotropic	20	9	10.4	9	16	14.6	12.7	14.8	11.9
Gaussian	30	11.3	12.5	11.1	22.3	21	18.3	21.3	17.7
	40	12.9	15.9	13.1	30.5	25.5	24.1	28	24.3

Table 3.16: Average errors of IDE methods for non-symmetrical datasets (undersampling) from table 3.15

	CD	GCD	MLE	DD	ANC	ESS	SSV1	SSV2
abs. error	13.31	11.01	12.67	5.858	7.671	10.03	5.929	7.908
rel. error	0.575	0.461	0.540	0.233	0.326	0.448	0.263	0.344

first note that the CD and MLE methods' tendency to underestimation becomes even worse for high values of m; to a lesser extent, we observe an increased underestimation for the GCD and an increased overestimation for the DD approach. On the other hand, the remaining methods (ANC, ESS, SSV1, SSV2) are less impaired by the effects of undersampling. Surprisingly, the average results of the ESS have minimally improved due to better estimates for the simplex dataset. Besides, the SSV1 yields an average relative error of 0.044, only slightly larger than for the initial test (0.031) with 10000 points. Thus, our SSV1 method proves to be quite robust against the influence of undersampling.

When it comes to the non-symmetrical datasets (compare tables 3.15 and 3.16), we basically see a similar trend. For the ellipsoid, rectangle and Gaussian datasets, the general tendency to underestimation worsens for all methods, where our SSV1 method again achieves better results than ANC and ESS. As before, the competitive performance of the DD method is a consequence of its overestimation issue canceling out with underestimation effects. Finally, the non-uniformly sampled paraboloid poses serious problems for most methods, i.e., CD, GCD, MLE, DD, and ESS, as their estimates allow for no meaningful distinction between the different ID values m. The ANC, SSV1, and SSV2 methods perform somewhat better, at least yielding increasing estimates for lower values of m.

In summary, our experiments show that ANC, ESS and SSV1 yield favorable results for undersampled datasets when compared to the remaining tested methods. Furthermore, specifically for anisotropic datasets, the SSV1 method's estimates are more reliable than those of ANC and ESS. Nevertheless and as expected, none of the approaches considered here is able to produce precise output values, especially for higher IDs, due to the undersampling.

Noise

Next, we evaluate the effects of noise on the different IDE methods. To begin with, let us remark that we do not expect our SSV methods to perform particularly well for noisy data. In subsection 3.4.4 we already showed that, given data of intrinsic dimension m embedded in $\mathbb{R}^D$ and tainted with D-dimensional Gaussian noise of $\sigma = 0.05$, the empirical average volume of s-dimensional simplices is much larger than the theoretically expected value for the noisefree case, especially for s approaching m. Since the SSV1 method analyzes only m-dimensional volumes when identifying the intrinsic dimension m, it is likely to be even more susceptible to overestimation of noisy datasets than the SSV2 method.

For the following noisy data comparison of all IDE methods we slightly modify our test settings in order to capture the interplay of different parameter values of the intrinsic dimension m, the ambient dimension D, and the standard deviation σ of the Gaussian noise. There are four series of experiments, one for each distinct value of $m = 2, 4, 10, 20$. The ambient dimension is selected as $D \in \{3, 6, 10, 20, 30, 50, 100\}$ with the natural

constraint $D > m$. Finally, we choose two noise levels as $\sigma_1 = 0.01$ for low noise and $\sigma_2 = 0.05$ for moderately strong noise. For each of those parameter combinations, we performed experiments for all symmetrical and non-symmetrical datasets introduced above. From this multitude of results, we selected the examples revealing the most important trends. Those are the m-ball as our reference object, the m-simplex as the most intricate object, and finally the isotropic m-dimensional Gaussian which stands out due to the fact that the noise is also of Gaussian nature. The corresponding results are presented in table 3.17 for $m = 2$, table 3.18 for $m = 4$, and table 3.19 for $m = 10, 20$.

At first, let us focus on the results for low noise level ($\sigma = 0.01$) and low ID values $m = 2, 4$. When comparing the different approaches, the CD method clearly outperforms all competitors, since it is the only one with (virtually) perfect estimation results for all test cases. The remaining methods rather seem to scale to a greater or lesser extent with the ambient dimension D, at least for the 2-ball, the 2-simplex and the 4-simplex. Apart from the CD method, both the MLE and the ESS methods often yield some slightly superior results than the rest of the field. Our SSV approaches on the other hand suffer from overestimation issues especially for $m = 2$ and $D \geq 20$. However, the problems of serious overestimation are also shared by the GCD, DD, and ANC methods, just to a somewhat lesser degree. Obviously, the noisy Gaussian dataset causes much fewer difficulties than the ball, while the simplex dataset remains a big challenge for most methods.

When it comes to a high level of noise ($\sigma = 0.05$), all methods, except for CD, yield nearly similarly poor results which usually scale with D. The overall performance of all methods is slightly better for $m = 4$ than for $m = 2$.

A possible explanation for the outstanding performance of the Grassberger-Procaccia approach (CD) is the fact that it is the only method completely relying on a multiscale scheme which includes measurements ranging from very small neighborhoods to the entire dataset. Since noise rather affects local than global patterns, a multiscale scheme benefits from the comparison of local and global information. On the contrary, the majority of our tested methods (MLE, DD, ANC, ESS, SSV1, and SSV2) rely on measurements within small neighborhoods, which are heavily disturbed by Gaussian noise. In fact, further experiments (not presented here) confirm that the ESS method profits from its relatively large number of nearest neighbors $k = 50$, while its estimates for noisy data deteriorate for smaller values of k.

Finally, we consider the estimation outcomes for higher IDs $m = 10, 20$ presented in table 3.19. For the m-ball and the m-Gaussian with $\sigma = 0.01$, ANC, ESS, SSV1, and SSV2 yield fairly reliable estimation results, while the remaining methods do not seem to suffer from noise issues, but their general tendency to underestimation (CD, GCD, MLE) or overestimation (DD), at least for $m = 20$. The m-simplex is again a greater challenge and only the CD method provides relatively precise estimates for $m = 10$. When considering the higher noise level ($\sigma = 0.05$), naturally, estimation results become more unreliable. Again, the Gaussian still allows for comparably good estimates of all methods, while the noisy ball and especially the simplex are prone to overestimation.

Table 3.17: IDE results for datasets with fixed ID $m = 2$, embedded in $\mathbb{R}^D$ and tainted with D-dimensional Gaussian noise: 2-ball, 2-simplex and 2-dim. Gaussian

	ID $m = 2$													
ball	Gaussian noise, $\sigma = 0.01$							Gaussian noise, $\sigma = 0.05$						
D	3	6	10	20	30	50	100	3	6	10	20	30	50	100
CD	2.2	2	2	2	2	2	2	3	5.6	8.6	11.4	5.2	2	2
GCD	3	5	7	11	14.2	20	30	3	5.8	8.2	14	18	25.2	38.4
MLE	2	4	5	8	10	14	22	3	6	8	14	18	25	38
DD	3	4	6	11	16	27.2	58	3	7	11	22	32	52.8	103
ANC	3	4	7	11	15	22.4	38.6	3	6	10	17.4	24.8	38.4	76.2
ESS	2	3	4	6	9	14	30.6	3	5	9	16.2	24	39	76
SSV1	2	5	8	16	23	38	>63	3	5.4	9	18.4	28	46	>63
SSV2	2	4	7	11	16.2	29	32	3	5	9	16	18	29	63.4
simplex	Gaussian noise, $\sigma = 0.01$							Gaussian noise, $\sigma = 0.05$						
D	3	6	10	20	30	50	100	3	6	10	20	30	50	100
CD	3	2.6	2	2	2	2	2	3	5.8	8.8	14.8	19.4	25.4	36.2
GCD	3	5	8	12.4	16	22.2	33.4	3	6	9	14.2	19	26.8	40.8
MLE	3	5	7	11	14	19	30	3	6	9	15	19.4	27	41.8
DD	3	6	9	17	26	42.2	84.6	4	7	12	23	34.2	56.2	110
ANC	3	5.6	8	14.2	19.8	29.4	52.8	3	6	10	19.2	27.4	44	84.4
ESS	3	4	6	11	17	28	56	3	6	9	18	26.2	43	83.8
SSV1	3	5	9	18	26	44	>63	3	6	9	19	28	47	>63
SSV2	3	5	9	14.2	21.8	23	48.6	3	5	8.8	13.6	19	30.8	69.2
Gaussian	Gaussian noise, $\sigma = 0.01$							Gaussian noise, $\sigma = 0.05$						
D	3	6	10	20	30	50	100	3	6	10	20	30	50	100
CD	2	2	2	2	2	2	2	3	5.4	4.4	2	2	2	2
GCD	2	4	5	8	10	14	22	3	5	8	13	16	23	34.4
MLE	2	3	3	4	6	8	12	3	5	7	11	14.4	20	31
DD	2	3	3	5	6	10	20.2	3	6	9	18	26	42.6	83.8
ANC	2	3	4	6	8	11.2	19	3	5.8	8	14.2	19.8	30	53.2
ESS	2	2	3	3	4	5	9	3	4	7	12	18	29	57
SSV1	2	3	5	11.2	16	27	53.8	3	5	9	18	27	45	>63
SSV2	2	3	4	7.8	10.8	13	25.6	3	5	9	14.2	21	22.8	46

Table 3.18: IDE results for datasets with fixed ID $m = 4$, embedded in $\mathbb{R}^D$ and tainted with D-dimensional Gaussian noise: 4-ball, 4-simplex and 4-dim. Gaussian

	ID $m = 4$											
ball	Gaussian noise, $\sigma = 0.01$						Gaussian noise, $\sigma = 0.05$					
D	6	10	20	30	50	100	6	10	20	30	50	100
CD	4	4	4	4	4	4	5.2	8	4	6.2	4	4
GCD	4	4.6	5.2	6	8	12.2	5	7.8	12	16	21.8	33.6
MLE	4	4	4	5	5	7	5	7	10	13	19	29
DD	5	5	6	6	7	9	6	9	15	22	36	71.8
ANC	4	4	5	6	7	9	5.2	8	13.8	18.4	28.6	53.6
ESS	4	4	4	4	5	6	5	6	10	14	23	46.8
SSV1	4	4	5	7	9	16	5	8	16	24.2	41	>63
SSV2	4	4	5	6	7	10.2	5	8	13	19	32.6	42.6
simplex	Gaussian noise, $\sigma = 0.01$						Gaussian noise, $\sigma = 0.05$					
D	6	10	20	30	50	100	6	10	20	30	50	100
CD	4	4	4	3.6	3.8	3.6	6	8.6	15.2	19.6	26	39
GCD	5	6	9.8	12	17	26.2	6	8.8	14.2	19	26.2	40.4
MLE	4	5	7	8	11	18	6	9	14	19	26	40.2
DD	5	6	9	11.2	17.2	35	7	12	22	32.6	53	104
ANC	4	6	9	11.4	16.8	29.6	6	10	18.6	26.2	41	83.4
ESS	4	5	6	7	10	19.4	6	9	17	24.2	40	78.2
SSV1	4.2	7	12	17.2	29	61.2	6	9	18	28	46	>63
SSV2	4	6	9.4	11	18	41.4	5	9	16	18.2	29	65.6
Gaussian	Gaussian noise, $\sigma = 0.01$						Gaussian noise, $\sigma = 0.05$					
D	6	10	20	30	50	100	6	10	20	30	50	100
CD	4	4	4	4	4	4	4	4	4	4	4	4
GCD	4	4	4	4	5	5	4	5	8	9.6	13	21
MLE	4	4	4	4	4	5	4	5	6	7	9	14
DD	5	5	5	5	5	6	5	6	7	9	13	23.2
ANC	4	4	4	4	4.2	5	4	5.4	7	9	12.6	21.2
ESS	4	4	4	4	4	4	4	5	5	6	8	14
SSV1	4	4	4	4	4	6	4	6	10	15	26	58.8
SSV2	4	4	4	4	4	6	4	6	8	10	12	23.6

Table 3.19: IDE results for datasets with fixed ID $m = 10$ and $m = 20$, respectively, embedded in $\mathbb{R}^D$ and tainted with D-dimensional Gaussian noise: m-ball, m-simplex and m-dim. Gaussian

	ID $m = 10$								ID $m = 20$					
ball	$\sigma = 0.01$				$\sigma = 0.05$				$\sigma = 0.01$			$\sigma = 0.05$		
D	20	30	50	100	20	30	50	100	30	50	100	30	50	100
CD	9	9	9.8	10	11.4	12.8	14.4	19	16.6	17.2	17.8	18.2	21.6	25.6
GCD	9	9.4	10	10	12	13.6	17.6	26.2	17	17.2	18	18.4	21.2	28.2
MLE	9	9	9	9.8	10	12	15	21	16	16	16	17	19.2	25
DD	14	14	14	15	16	19	25	41	29	29	30	31	36.8	51.4
ANC	10	10	10.2	11	12.6	15	20.2	33.4	20	20	21	22.8	27	39.6
ESS	10	10	10	10	11	13	16	26	20	20	20	21	25	34
SSV1	10	10	11	12	14	19	29	60.4	20	21	22	24	33	60
SSV2	10	10	10	11	12	15	20	38	21	21	22	23	28.2	36
simplex	$\sigma = 0.01$				$\sigma = 0.05$				$\sigma = 0.01$			$\sigma = 0.05$		
D	20	30	50	100	20	30	50	100	30	50	100	30	50	100
CD	9.8	11.2	9.2	9	15.6	20	27.4	42.6	16.8	19.2	24.2	21.2	29.6	46.2
GCD	10.2	12	14.6	21	15	19.8	27	41.4	16.2	19	25.8	20.4	28.2	43
MLE	9	10	12	16	15	20	28	43	15	17	22	22	30	46.2
DD	13	15	18.2	28	23	33.2	54.6	108	25	30	44	35.8	59	118
ANC	11	12	15	22.6	19.2	27.4	44.6	89	19.2	23.8	35	30	48.8	97.2
ESS	10	11	13	18.6	17.8	26	42	84	19	22	32	29	47.2	93.8
SSV1	12	16	23	46.2	19	28	46.8	>63	23	32	60	28	47.8	>63
SSV2	11	12	15	24.2	16.8	19	30	67.8	18	23	38	20	32	74.6
Gaussian	$\sigma = 0.01$				$\sigma = 0.05$				$\sigma = 0.01$			$\sigma = 0.05$		
D	20	30	50	100	20	30	50	100	30	50	100	30	50	100
CD	9	9	9	9.2	9.2	9.6	10.2	10.4	16	16.2	16	16.2	16.8	17
GCD	9	9	9	9	9	10	10	11	16	16	16	16	16	16.2
MLE	10	10	10	10	10	10	10	11	17	17	17	17	17	17
DD	13	13	12.8	13	13	13	13.8	15	25	25	25	25	25.2	26
ANC	11	11	11	11	11	11.4	12	13	21.6	22	21.4	21.6	22.4	23
ESS	10	10	10	10	10	10	11	12	20	20	20	20	20	21
SSV1	9	9.8	10	10	10	11	12	15.6	19	19	19	19.4	20	22
SSV2	9.8	10	10	10	10	10	11	12	17.4	17.6	18	18	18	19

In summary, we observe that the CD method, due to its multiscale approach, clearly outperforms the remaining methods when it comes to our test scenarios featuring noisy datasets. However, recalling the results for the m-dimensional paraboloid (see table 3.11), the CD method is obviously not able to recover the true high-dimensional structure of this object, since its estimate is between 2 and 3 for all values $m = 4, 10, 16, 20, 30, 40$. The dilemma of separating noise from structure in high-dimensional data has of course no universal solution.

Our SSV methods based on local simplex volumes are not particularly well-suited for noisy data. Especially the SSV1 approach is inclined to highly overestimate the ID of noisy datasets, while the SSV2 method is often on the same level as the ANC or GCD method. However, the above results also show that *all* tested methods (except for CD) are sensitive to noise in a quite comparable way. Considering the vast discrepancies between the results for the m-simplex and the m-Gaussian, it is justified to conclude that a combination of a more complex low-dimensional structure with mild noise can make estimation impossible for all practical purposes. As a more promising strategy for data tainted with noise we suggest to apply a denoising (i.e., noise reduction) algorithm before utilizing any IDE method.

Datasets with high curvature proposed in [HA05]

The next series of experiments is based on datasets presented in [HA05] that have also been re-used in the IDE review article and benchmark proposal in [CCCR15], which is why we include them here. The descriptions of the appropriate datasets can be found in table 3.20, while for the exact definitions, we refer to the original publication. The numerical estimation results, each for two numbers of sample points $N = 100$ and $N = 10000$, are given in table 3.21. We omitted the sets "GCD1", "GCD9", "GCD10", and "GCD12" representing m-dimensional spheres, cubes, and Gaussians, respectively, since they have already been analyzed in our first experiment.

The experiments with $N = 10000$ points first show that, except for the DD method's tendency to overestimation, all methods produce reliable and correct results for nearly all datasets, except for "GCD6" and "GCD8".

The "GCD8" dataset is defined via the following function:

$$
\begin{aligned}
\boldsymbol{X}(\boldsymbol{y}) &: [0,1]^{12} \to \mathbb{R}^{72}, \\
X_{2i-1}(\boldsymbol{y}) &= y_{i+1}\cos(2\pi y_i), & X_{2i}(\boldsymbol{y}) &= y_{i+1}\sin(2\pi y_i), \quad (\forall i = 1,\dots,11), \\
X_{23}(\boldsymbol{y}) &= y_1\cos(2\pi y_{12}), & X_{24}(\boldsymbol{y}) &= y_1\sin(2\pi y_{12}), \\
X_{j+24}(\boldsymbol{y}) &= X_{j+48}(\boldsymbol{y}) = X_j(\boldsymbol{y}), \quad (\forall j = 1,\dots,24).
\end{aligned}
$$

Thus, it is a 12-dimensional manifold embedded in a 24-dimensional subspace, where the 24 coordinates are duplicated two times to yield 72 dimensions in total. In [HA05], the authors remark that their sampling procedure gives rise to a non-uniform probability measure usually resulting in underestimation, while, at the same time, the high curvature

Table 3.20: GCD dataset descriptions (compare also [HA05])

dataset	D	m	description
GCD0	3	1	highly oscillating sinusoid (see eq. (3.123) and fig. 3.6)
GCD2	5	3	3D affine space embedded in $\mathbb{R}^5$
GCD3	6	4	4D manifold (mixture of sin, cos, x^2) in $\mathbb{R}^6$
GCD4	8	4	4D manifold (mixture of sin, cos) in $\mathbb{R}^8$
GCD5	3	2	2D helix in $\mathbb{R}^3$
GCD6	36	6	6D "strange" manifold in $\mathbb{R}^{36}$ (see text)
GCD7	3	2	swiss role (see eq. (2.24) and fig. 3.5)
GCD8	72	12	12D "strange" manifold in $\mathbb{R}^{72}$ (see text)
GCD11	3	2	10-times twisted Möbius strip in $\mathbb{R}^3$ (see fig. 3.5)
GCD13	D	1	1-dim. curve in $\mathbb{R}^D$

Table 3.21: Numerical IDE results for GCD datasets

dataset	D	m	N	CD	GCD	MLE	DD	ANC	ESS	SSV1	SSV2
GCD0	3	1	100	1	2	1	1	1.8	2	1	1
			10000	1	1	1	2	1	2	1	1
GCD2	5	3	100	2.3	3.1	2.9	3	2.9	3	2.9	2.9
			10000	3	3	3	4	3	3	3	3
GCD3	6	4	100	3	3.6	3.4	4.3	3.3	3.8	4	4
			10000	4	4	4	5	4	4	4	4
GCD4	8	4	100	4	4	5	6.4	4.6	7	6.1	6
			10000	4	4	4	5	4	4	4	4
GCD5	3	2	100	2	2.1	2	2.7	2	3	2	2
			10000	2	2	2	2	2	2	2	2
GCD6	36	6	100	5.9	5.9	6.9	10.9	8.4	11	10	9.1
			10000	6	6	6	9	6.2	7	8	7
GCD7	3	2	100	2	2	3	3	2	3	2.3	2.3
			10000	2	2	2	2	2	2	2	2
GCD8	72	12	100	10.4	11.9	11.5	25.3	19.6	23.1	22	19
			10000	12	12	14	22.1	16.1	18	21	17
GCD11	3	2	100	1.6	2	2	2	2	2	2	2
			10000	2	2	2	2	2	2	2	2
GCD13	5	1	100	1	1	1	2	1	2.9	1	1
			10000	1	1	1	1	1	1	1	1
GCD13	20	1	100	1.2	1	2.3	2.9	1.1	3.6	3	2.9
			10000	1	1	1	1	1	1	1	1

leads to overestimation. Looking at the results in table 3.21, the two effects seem to cancel out for the CD and the GCD approach, while the ANC, ESS, SSV1 and SSV2 methods yield outcomes between 16 and 22, which is close to the original embedding dimension of 24. The "GCD6" is in fact constructed in the exact same way than the "GCD8" data, just with lower dimensions, and thus, similar effects are considered for this dataset.

Summing up the results for highly curved datasets as proposed in [HA05], our two SSV approaches achieve accurate results comparable to those of the other methods, except for the DD method's poor performance. Naturally, a very high curvature can lead to overestimation, while a lower sampling number ($N = 100$) slightly deteriorates results in some cases.

3.5.5 Results from real-world data

In this subsection, we examine the performance of the selected IDE algorithms for data collected in real-world scenarios. The datasets originate from various different research areas such as meteorology, cartography, particle physics, image and voice recognition, car crash simulations, and the modeling of wine preferences.

As several datasets in fact correspond to (one-dimensional) time series embedded in some higher-dimensional ambient space, we first provide a compact introduction into the topic of time series analysis and attractor dimensions. Next, we describe the detailed characteristics of all chosen real-world datasets; for a quick overview we refer to table 3.22. Subsequently, the empirical results of all IDE methods are presented and discussed with a particular focus on the differences between synthetic and real-world data and the arising challenges.

Time series and estimation of attractor dimensions

Before we introduce the datasets representing discrete time series, we give a very short summary on the background of nonlinear dynamics methods and attractors following [CF09] in order to justify the application of intrinsic dimension estimators in this context. For deeper insights we refer to [Ott02, KS04, Sma05]. First, we assume that we have a time series $x(t_i)$ with equidistant time steps $t_1, t_2, \ldots, t_l$. The underlying model producing the time series is some (unknown) m-dimensional dynamical system, often represented by a set of differential equations. In order to reconstruct certain system properties and in particular in order to determine the unknown model order m, the authors of [PCFS80] first proposed the so-called *method of delays*. For this purpose, the time series is embedded into a higher-dimensional space by aggregating D consecutive elements as

$$\mathbf{X}(t_i) = \Big(x(t_i), x(t_{i+1}), \ldots, x(t_{i+D-1})\Big)^T, \qquad (i = 1, 2, \ldots). \tag{3.126}$$

Now, under some mild assumptions, the embedding theorem by Takens [Tak81] states that if $2s + 1 \leq D$, there exists a diffeomorphism between the manifold M generated by the points $\mathbf{X}(t_i)$ and the attractor U of the underlying dynamical system, where s is the dimension of U. Here, the so-called *attractor* is a set of numerical values toward which the associated system tends to evolve, generally almost independently of the starting parameters. Attractors can be either singular points as well as curves, high-dimensional manifolds, or fractal sets commonly known as *strange attractors*. The latter can be described by fractal dimensions and have essentially motivated the concept of the correlation dimension [GP83].

In practice, after selecting a sufficiently high embedding dimension D, an IDE method can be utilized to estimate the intrinsic dimension of the point set $\mathbf{X}(t_i) \subset \mathbb{R}^D$, i.e., the dimensionality of the system attractor, which in turn allows to draw conclusions about the corresponding model order.

Description of the test datasets

The three datasets CHUA-DSVC1, CHUA-DSIL and CHUA-SPIVC1 have been presented in [ARM97]. They can be accessed at http://www.cpdee.ufmg.br/~MACSIN/services/data/data.htm and correspond to measurements of a specific hardware realization of Chua's circuit. This simple electronic circuit, first introduced by Chua in [CKM85] and [CKM86], features a chaotic behavior and nonlinear dynamics and has therefore become very popular for the demonstration of distinctive effects of chaos theory. The precise measurements are the inductor current for the so-called double-scroll attractor (DSIL), the voltage across some capacitor in the double-scroll attractor (DSVC1), and the voltage across some capacitor in the so-called spiral attractor (SPIVC1). The original datasets consist of 5000 (DSVC1), 15000 (DSIL) and 15000 (SPIVC1) points and have been embedded in 20-dimensional space using the method of delays described above. According to [ARM97] , the estimated Lyapunov dimensions of the attractors associated with the three time series are between 2.24 and 2.26, while the estimated correlation dimensions vary between 1.887 and 1.957. Furthermore, the authors remark that the two larger datasets (DSIL and SPIVC1) contain significantly more noise than the other one, which is why we rather expect estimates of 3 or above.

The two datasets CLIM-STOC and CLIM-TENE represent time series of measured daily average temperatures, represented in 10^{th} of degrees Celsius. The data has been described in [KTWK+02] and can be accessed at the large archive of the *European Climate Assessment and Dataset* [ECA16]. We selected two stations at Stockholm (station ID 10, daily data from January 1 1756 to December 31 1999, i.e., 89119 samples) and Santa Cruz de Tenerife (station ID 3959, daily data from April 1 1964 to April 30 2013, i.e., 17927 samples), which are so-called "non-blended" data, meaning that the temperature values originate from a single fixed station. The time periods have been chosen in order to avoid any missing values present in the original datasets. As the underlying models of most climatic variables must be presumed to be of very high complexity, the

Table 3.22: Real world dataset descriptions

dataset	N	D	type	description	source
CHUA-DSVC1	4981	20	real	measurements from hardware realization of Chua's Circuit: inductor current (IL) and voltage across capacitor (VC1)	[ARM97]
CHUA-DSIL	14981	20	real		
CHUA-SPIVC1	14981	20	real		
CLIM-STOC	89100	20	real	measured daily average temperatures at stations in Stockholm and Santa Cruz de Tenerife; two embedding dimensions $D_1 = 20$ and $D_2 = 50$	[KTWK$^+$02]
CLIM-STOC	89070	50	real		
CLIM-TENE	17908	20	real		
CLIM-TENE	17878	50	real		
COVTYPE	581012	55	int.	cartographic variables (e.g. elevation, slope) used to predict forest cover type	[BD99]
HEPMASS	3500000	27	real	features of simulated particle collisions	[BCF$^+$16]
ISOLET	7797	617	real	acoustic features of recorded samples of spoken letters of the English alphabet	[CF91]
ISOMAP-FACE	698	4096	real	64×64 image pixels of 3D head rendered with different poses and lighting	[TdSL00]
MNIST-0	6903	784	int.	28×28 image pixels of hand-written digits 0, 1, and 9	[LBBH98]
MNIST-1	7877	784	int.		
MNIST-9	6958	784	int.		
POKER	997872	11	int.	suit and rank of five playing cards drawn from a standard deck of 52	[COD02]
SANTA-D	99981	20	real	time series generated by numerical integration of equations of motion	[WG94]
SANTA-D	99951	50	real		
TAURUS-B1	273	4539	real	displacement vectors of finite element nodes of a vehicle body model for 273 different crash test simulations	[BGG16]
TAURUS-B8	273	13998	real		
TAURUS-ALL	273	86712	real		
WINE-WHITE	3961	12	real	physicochemical attributes of white wine (Portuguese "Vinho Verde")	[CCA$^+$09]

corresponding embedding dimension D for the method of delays should be chosen sufficiently large. Since larger values of D might also reveal more details in the data, we decide to evaluate our experiments with the two different choices $D_1 = 20$ and $D_2 = 50$ to compare the respective outcomes.

The COVTYPE dataset, accessible via the UCI machine learning repository [Lic17] and described in [BD99], features a list of several cartographic variables with the purpose of predicting one of seven different forest cover type classes. The first 10 attributes are: elevation, aspect, slope, horizontal distance to hydrology, vertical distance to hydrology, horizontal distance to roadways, hillshade 9am, hillshade noon, hillshade 3pm, horizontal distance to fire points. These quantitative measurements are given as integer values in their respective units, such as meters, azimuth or degrees. The following 44 binary attributes are of qualitative nature and represent the absence (0) or presence (1) of certain wilderness areas (4 attributes) and soil types (40 attributes). Finally, the last attribute denotes the corresponding class of the forest cover type. The intrinsic dimensionality of this dataset is unknown.

In high-energy physics experiments, machine learning techniques are used to facilitate the search for signatures of exotic or new, hypothetical particles. The HEPMASS dataset presented in [BCF+16] (available at the UCI MLR) has been generated by millions of simulated particle collisions using the simulation interface "MadGraph 5", see [AHM+11]. The goal here is the training of a neural network for the purpose of separating particle-producing collisions from a background source. The particularity of the approach in [BCF+16] is the use of additional high-level features to train the neural network, as opposed to classical methods completely relying on low-level features. Hence, the data consists of 27 normalized attributes, including 22 low-level features (particle momenta, number of jets, etc.) and 5 high-level features (masses of intermediate objects). The authors provide several datasets, from which we select the test set of the dataset called "1000" with fixed mass. It consists of 3.5 million points with 27 features; also, the ID of this dataset has not been examined before and is unknown.

The ISOLET dataset, available at the UCI MLR and described in [CFM90, CF91], consists of recorded samples of all letters of the English alphabet spoken in an isolated test environment. Each letter was spoken two times by 150 native English speakers, resulting in a total of $52 \cdot 150 = 7800$ samples; due to recording problems, only 7797 samples are present in the dataset. Furthermore, for training and testing, the data has been divided into five groups of 30 speakers each, referred to as ISOLET1, ISOLET2, ..., ISOLET5. For each utterance, 617 features were computed, that can be classified into one of four groups: contour features, sonorant features, pre-sonorant and post-sonorant features. The ISOLET data has been analyzed e.g. in [KLN+10], where the authors compute scale-dependent correlation dimensions on different scales using 100 measuring points and generate a plot of dimensionalities against scales. Here, the dimensionality explodes for small scales, which is probably due to a high level of noise in the acoustic measurements. Although it is hard to detect any distinctive linear behavior of the examined curve, the authors propose a rough estimate for the intrinsic dimension as

$\hat{m} \approx 13$.

For examining their ISOMAP dimensionality reduction algorithm, the authors of [TdSL00] introduced the ISOMAP-FACE dataset that has become popular in the field of machine learning and is currently available at http://web.mit.edu/cocosci/isomap/datasets.html. The data consists of 698 instances of 64×64 gray-level image pixels, each image representing the same three-dimensional head of a statue rendered with different poses and lighting directions. This data is generally considered to have an intrinsic dimension of $m = 3$, where the three degrees of freedom correspond to the up-down pose, the left-right pose, and the lighting direction.

The MNIST database contains 70000 gray-level 28×28 pixel images of handwritten digits and is widely used in the field of image classification, e.g. via neural networks with promising results in [CMS12]. It has first been investigated in [LBBH98] and is available at http://yann.lecun.com/exdb/mnist/. The pictures (produced by 500 writers) are grouped according to the represented digits $(0, 1, \ldots, 9)$. We utilized the combined training and test datasets MNIST-0 with 6903 samples, MNIST-1 with 7877 samples, and MNIST-9 with 6958 samples. Even though some publications ([CH04b, HA05, CCB$^+$14]) seek to assign a fixed intrinsic dimension to each distinct sets of images, we believe that a scale-dependent dimension estimate is much more appropriate to capture the intrinsic structure of this dataset due to the variety of different handwriting styles.

The POKER dataset (see [COD02]) has been generated to test the performance of a certain rule induction algorithm named "RAGA" [COD99] relying on genetic programming and is accessible via the UCI MLR. Each record is the exact description of a random sample of five playing cards out of 52 using the two categories *suit* (hearts, spades, diamonds, clubs) and *rank* (2, 3, ..., queen, king, ace) resulting in 10 different integer attributes. The last attribute assigns one out of 10 different classes to each sample, representing the *poker hand*, i.e., the categorization of the current set according to the rules of the poker game. After removal of duplicates, we obtain a dataset of 997872 points of dimension 11. Since all samples have been generated randomly and all 10 attributes of the five cards are relevant for the classification, the intrinsic dimension should be equal $m = 10$.

Several interesting datasets originate from the so-called "Santa Fe Time Series Competition" initiated by Gershenfeld and Weigend in 1991, see [WG94]. The original webpage http://www-psych.stanford.edu/~andreas/Time-Series/SantaFe.html is offline, but past versions and also the datasets can still be accessed using the *Internet Archive* at http://archive.org. The SANTA-D time series, consisting of 100000 points, has been generated by numerical integration of the equations of motion for a damped, driven particle. Here, a fixed-step 4th order Runge-Kutta routine was used, and the underlying model has 9 degrees of freedom. In order to analyze the influence of the embedding dimension, we again select two different values $D_1 = 20$ and $D_2 = 50$ for the method of delays. The ID of the SANTA-D dataset is of course $m = 9$.

The TAURUS data has been generated in the context of crash test simulations of a Ford Taurus car, see [BGG16]. The original model of the vehicle body consists of

approximately 900000 finite element nodes and has been built by the *National Crash Analysis Center*, http://www.ncac.gwu.edu/. In the experiments described in [BGG16], multiple crash simulations are performed where 19 parameters (basically the material thickness of certain components) have been varied leading to a total number of 273 different simulations. The most crucial components of the vehicle body are the so-called "beams" absorbing the main impact energy of a frontal crash. Here, we are dealing with 15 beams consisting of 28904 finite element nodes. The critical information are now the displacement vectors of each node between two fixed time steps before and after the crash. Hence, for each of the $N = 273$ simulations, there are $D = 3 \cdot 28904 = 86712$ entries corresponding to the components of all displacement vectors. According to the authors, when using a principal manifold learning technique, only two degrees of freedom are sufficient to capture most of the simulation effects relevant in practice for a single beam. We consider three variants: TAURUS-B1 ($D = 4539$) and TAURUS-B8 ($D = 13998$) correspond to a solitary selected beam, respectively, while TAURUS-ALL ($D = 86712$) represents the complete dataset with all 15 beams.

The WINE-WHITE dataset is described in [CCA+09] and also available at the UCI Machine Learning Repository. Here, a total number of 3961 variants (after duplicate removal) of the Portuguese "vinho verde" white wine has been analyzed with respect to certain quantitative physicochemical attributes, such as acidity, residual sugar, chlorides, sulfates, alcohol and others. The latter sum up to 11 different features, while the last component represents a (subjective) measure of quality, i.e., a class variable between 0 and 10 chosen by test subjects. In [CCA+09], the authors use a support vector machine approach to predict the taste preferences and state that "most of the physicochemical tests used are relevant". However, the intrinsic dimensionality of this wine dataset is not examined here, which is why it is interesting to compare our IDE methods' performances.

Analysis of the IDE results

The empirical results of our selected IDE methods can be found in table 3.5.5. We divide our pool of datasets into two groups: the first group consists of all datasets where either the true intrinsic dimension m is known a priori, or the estimated ID of all compared methods turned out to be low a posteriori. The second group contains datasets that do not have a known unique ID value; the reason can either be a high level of noise or their intrinsic structure that can not be captured properly via some specific low-dimensional manifold. In the latter case, generally, the data appears to have a different intrinsic dimension when analyzed on different scales, and there is usually no perfect answer to the question of a unique ID m. Consequently, distinct approaches lead to different results that require careful interpretation and further considerations.

For the CHUA datasets, we already observe the difference between the less noisy CHUA-DSVC1 data and the other two measurements tainted with higher levels of noise. While for the first dataset, nearly all methods are able to detect the true underlying dimension of $m \approx 2$, the noise in the CHUA-DSIL and CHUA-SPIVC1 data lead to a

Table 3.23: Numerical IDE results for real-world datasets

dataset	N	D	m	CD	GCD	MLE	DD	ANC	ESS	SSV1	SSV2
CHUA-DSVC1	4981	20	2	2	3	2	3	3	2	2	2
CHUA-DSIL	14981	20	2	2	10	8	11	10	7	17	11
CHUA-SPIVC1	14981	20	2	1	9	7	9	9	5	14	11
COVTYPE	581012	55	?	5	5	3	4	3	3	3	3
ISOMAP-FACE	698	4096	3	3	3	4	8	4	7	9	6
POKER	997872	11	10	9	10	10	13	1[a]	10	10	10
SANTA-D	99981	20	9	3	7	7	9	8	7	9	8
SANTA-D	99951	50	9	7	7	7	10	8	8	12	9
TAURUS-B1	273	4539	?	6	5	5	6	6	5	6	5
TAURUS-B8	273	13998	?	7	8	8	11	10	9	14	11
TAURUS-ALL	273	86712	?	9	9	9	14	12[b]	11	20	12
WINE-WHITE	3961	12	?	3	4	4	4	4	3	4	4
CLIM-STOC	89100	20	?	12	9	14	17	16	14	17	13
CLIM-STOC	89070	50	?	13	11	16	26	17	22	36	21
CLIM-TENE	17908	20	?	12	11	14	17	17	15	18	14
CLIM-TENE	17878	50	?	14	14	17	28	20	25	40	25
HEPMASS	3500000	27	?	13	12	14	20	15[c]	16	19	15
ISOLET	7797	617	?	13	14	17	42	18	34	>63	41
MNIST-0	6903	784	?	14	12	14	25	18	19	51	22
MNIST-1	7877	784	?	7	8	11	15	13	12	28	14
MNIST-9	6958	784	?	11	12	14	24	16	19	45	21

[a] For the POKER dataset, the number of nearest neighbors $k = 10$ seems to be too small resulting in a failure of the ANC method to estimate the corresponding ID. The outcomes for $k = 20$ and $k = 30$ are given by $\hat{m}_{\mathrm{ANC}} = 8$ and $\hat{m}_{\mathrm{ANC}} = 7$, respectively.

[b] Since the dimension $D = 86712$ of the TAURUS-ALL data was too large for the ANC `Matlab` routine, it has been projected into $\mathbb{R}^{273}$ using multidimensional scaling (MDS).

[c] The number of points of the HEPMASS dataset had to be reduced to $\tilde{N} = 500000$ via random sampling in order to make the associated `Matlab` routine work properly.

serious overestimation for all approaches except for the CD method. However, for the SPIVC1 data, the latter yields an underestimate of $\hat{m} = 1$, thus none of our tested methods succeeds to reveal the correct ID for this example.

The COVTYPE dataset obviously features a low-dimensional structure. Considering the estimation results, this is one of the rare cases where both the CD and GCD methods return a larger estimate $\hat{m} = 5$ than most of the remaining approaches with $\hat{m} = 3$. We will come back to this issue in the following subsection, where we evaluate the scale-dependent correlation dimension.

The ISOMAP-FACE data is considered to have an ID of $m = 3$ due to its three degrees of freedom (horizontal pose, vertical pose, and lighting direction). Nevertheless, half of our tested methods (DD, ESS, SSV1 and SSV2) yield estimates between 6 and 9, which suggests that there is some higher-dimensional structure on a certain scale.

The intrinsic dimension of the POKER data is correctly identified by almost all estimators, albeit with a slight underestimation (CD) or a moderate overestimation (DD). The ANC approach has some problems with its standard parameter for the number of nearest neighbors of $k = 10$. For this reason, additional tests have been performed for larger values $k = 20$ and $k = 30$; the associated estimates given by $\hat{m}_{\mathrm{ANC}} = 8$ and $\hat{m}_{\mathrm{ANC}} = 7$, respectively, are closer to the true ID, however still not quite accurate.

The ID of the SANTA-D time series dataset, generated by a physical model with 9 degrees of freedom, is generally slightly underestimated by most methods, for both embedding dimensions $D_1 = 20$ and $D_2 = 50$; only the DD and our SSV1 approach yield correct or slightly larger estimates in both cases.

Considering the TAURUS data, we in fact analyzed each of the 15 beams individually and selected beam B1 as an average one and B8 as the one with the generally largest estimation results of all methods. It is interesting to see that for the smaller B1 beam, there is a consensus amongst all estimators for an ID value of 5 or 6, while the variance in the outcomes is much larger for the other beam B8. Furthermore, the union of all beams only seems to have a slightly higher intrinsic dimension than the B8 beam.

Finally, for the WINE-WHITE data, no precise ID value is known; nevertheless, all tested techniques equally return a value between 3 and 4.

Next, we focus on the real-world datasets of the second group. When considering the results for the CLIM-STOC and CLIM-TENE time series data, we observe an interesting aspect. As opposed to the SANTA-D time series, where the choice of the embedding dimension has only a small effect on the estimation results, the estimates for the climate datasets vary to a larger extent with different values of D, especially for the methods DD, ESS, SSV1 and SSV2. We believe that the measurements of the daily mean temperatures at a fixed location for several decades are indeed dependent on many underlying variables. Even though the main structure might be representable by a number of 12 to 14 features, the embedding in higher-dimensional space can unveil further and more subtle patterns. This could explain the distinct behavior of the tested IDE methods for $D_2 = 50$.

The HEPMASS dataset consists of 3.5 million points in 27 dimensions. Since the ANC method's `Matlab` code would not process a dataset of such size, we reduced the number

of points to 500000 via random sampling, solely for this method. The results here span from values between 12 to 20. Due to the tendency of the CD and GCD methods to underestimate IDs and the tendency of the DD and SSV1 methods to overestimate IDs, we suppose that the "true" ID here might be a value close to $m = 15$ or $m = 16$.

For the ISOLET data, we find a large variance in the estimation results ranging between $\hat{m} = 13$ for the Grassberger-Procaccia (CD) approach, while our SSV1 method could not even detect an ID below 64, and DD, ESS and SSV2 return values of 34 and above. Similarly to the climate time series considered before, we assume that the voice recordings of the ISOLET data comprise high-dimensional structures on a very fine scale causing the higher estimates of certain methods. We also refer to the upcoming analysis of the scale-dependent correlation dimension.

The outcomes for the MNIST datasets exhibit similar characteristics, however here, the results of the ANC, ESS and our SSV2 approach are relatively close to each other. Although the estimates of the SSV1 are larger than those of the other methods, one clear trend is shared by the entire test group: the intrinsic complexity of the MNIST-1 data is rated lower than that of the MNIST-0 and MNIST-9 variants.

Summary of our methods' results

The characteristics of our IDE methods analyzed for synthetic datasets in subsection 3.5.4 are generally confirmed by our experiments with real-world data. For many datasets in the first group with low ID and no dominating noise (CHUA-DSVC1, COVTYPE, POKER, TAURUS-B1, WINE-WHITE), our SSV approaches perfectly agree with the results of the remaining test field. On the other hand, higher noise levels can render ID estimation nearly impossible (CHUA-DSIL, CHUA-SPIVC1). The effect of underestimation of the CD, GCD and MLE methods is again revealed for the SANTA-D data, where only two methods, DD and SSV1, do not suffer from this issue. Remember that in the context of dimension reduction, overestimation of the ID is much less of a problem than underestimation.

For the second group of datasets, the variances in the outcomes is often high which is why we suppose the intrinsic dimensionality is rather a scale-dependent quantity in most cases. The results of the ESS and SSV2 methods are often very close to each other, while the SSV1 approach yields much larger estimates than the rest in some cases (CLIM-TENE for $D = 50$, ISOLET, MNIST).

3.5.6 Further numerical evaluations

Analysis of the scale-dependent correlation dimension for selected datasets

In order to shed further light on some particularities of the above estimation results, we evaluate the scale-dependent correlation dimension for several datasets. We recall the definition of the correlation dimension (see 3.1.4, eq. (3.11)) for a given set $\mathcal{X} =$

$\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ as

$$D_{\text{cor}}(\mathcal{X}) = \lim_{r \to 0} \frac{\log C_r(\mathcal{X})}{\log r}, \tag{3.127}$$

where the corresponding correlation sum is defined as

$$C_r(\mathcal{X}) = \frac{2}{N(N-1)} \sum_{1 \leq i < j \leq N} H\left(r - \|\boldsymbol{x}_i - \boldsymbol{x}_j\|\right). \tag{3.128}$$

In practice, for a given scale $[r_1, r_2]$, the scale-dependent correlation dimension is defined as

$$\hat{m}_{\text{cor}}\left(r_2, r_1\right) = \frac{\log \frac{C_{r_2}(\mathcal{X})}{C_{r_1}(\mathcal{X})}}{\log \frac{r_2}{r_1}}. \tag{3.129}$$

We now apply the approach suggested in [KLN+10] to generate suitable function plots of the correlation dimension. For this purpose, the minimum and maximum of all distances of a given dataset, $\delta_{\min}$ and $\delta_{\max}$, are determined and the corresponding interval is then segmented as $[\delta_{\min} = r_0 < r_1 < \ldots < r_S = \delta_{\max}]$, where $S = 100$ and the r_k are distributed logarithmically. Since we choose a large number of 100 segments, it seems reasonable to smoothen the local estimates using a triangular kernel of a certain window length w. The final function we consider is therefore defined as

$$\hat{m}\left(r\right) = \frac{1}{2w} \sum_{\substack{j=i-w \\ j \neq i}}^{i+w} \hat{m}_{\text{cor}}\left(r_i, r_j\right), \qquad \text{for } r \in [r_i, r_{i+1}). \tag{3.130}$$

The corresponding function plots for several selected datasets are shown in figure 3.9.

First, let us emphasize that the scale-dependent correlation dimension can also involve misleading conclusions, which is why it must be interpreted with caution. To this end, recall that a viable estimator based on the correlation dimension usually seeks to find the largest region $[a, b]$ of almost constant slope in the plot, and then takes the local estimate $\lim_{r \to a} \hat{m}(r)$, since the correlation dimension is defined for the limit of $r \to 0$ (see also the corresponding paragraph in subsection 3.2.2). Now let us compare the first two plots in figure 3.9 for synthetic datasets, the 10-dimensional ball and the 40-dimensional simplex with their associated estimation results of the CD method, which are $\hat{m}_{\text{CD}} = 9.1$ and $\hat{m}_{\text{CD}} = 23.4$, respectively (see table 3.9). The two plots show a relatively similar curve. However, in the first case, the sharp peak at the left end seems to be due to some spurious small-scale effects leading to a considerate (local) overestimation, while in the second case, the peak in fact indicates the true underlying dimensionality. Consequently, while the strategy of the CD estimator is successful for the 10-ball, it fails for the 40-simplex. In contrast, our SSV1 method yields more accurate estimates in both cases, which are $\hat{m}_{\text{SSV1}} = 10$ and $\hat{m}_{\text{SSV1}} = 37$, respectively.

In spite of this deficiency, the scale-dependent correlation dimension is nevertheless a simple and established way to reveal particular structures on different scales in the

data. Consider e.g. the plot for the WINE-WHITE dataset in figure 3.9. The global maximum is very close to 4 and — in contrast to many noisy datasets — no increase in dimensionality can be found for $r \to 0$, which explains the quite unanimous results of all tested IDE methods (compare table 3.5.5).

The curve for the COVTYPE data is below the value of 5 except for very small scales. Note here that the quotient of the maximum and minimum distance $\delta_{\max}/\delta_{\min} = 3.13 \cdot 10^3$ is large compared to most other datasets. We believe that this is the reason why most methods yield an estimate of $\hat{m} = 3$ rather than higher values.

The different estimation results for the ISOMAP-FACE data ($\hat{m} \in \{3, 4\}$ for CD, GCD, MLE, ANC, and $\hat{m} \in \{7, 8\}$ for DD, ESS, SSV1, SSV2) are probably reflected in the spike visible at the left end of the corresponding function plot.

The HEPMASS data curve clearly features an interval of nearly constant slope with only a narrow spike for small scales r leading to relatively close estimation results of our tested methods ranging from 12 to 20.

Finally, let us consider the scale-dependent correlation dimension for the CLIM-STOC and the ISOLET datasets, two examples with a high variation in the estimation results. It is remarkable that the associated estimates of the SSV1 method, $\hat{m}_{\text{SSV1}} = 36$ and $\hat{m}_{\text{SSV1}} > 63$, are higher than the respective global maxima in the plots. We believe that the analysis of high-dimensional simplex volumes is able to unveil certain high-dimensional structures on very small scales, that might not be easily identifiable by the pure exploration of Euclidean distances. Naturally, the important question remains open whether those high-dimensional structures represent valuable information or rather unwanted artefacts (noise). In general, this question can not be answered by an automatic scheme without further knowledge of the dataset characteristics.

3.5.7 Runtime evaluations

In this subsection, we present some runtime measurements of our methods. For this purpose, let us first recall the cost complexities of our two approaches, that have already been presented in subsection 3.4.6. For a given dataset $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\} \in \mathbb{R}^D$ of intrinsic dimension m, the associated complexities evaluate as

$$\text{SSV1:} \quad \sum_{d=1}^{\hat{m}} \left[\mathcal{O}\left(N \cdot D \cdot (k_d \cdot \log N + k_d/\delta + 1/\delta^3 + k_d^2) + N \cdot C \cdot d^3 \right) \right],$$

$$\text{SSV2:} \quad \sum_{d=1}^{s_{\max}^{(2)}} \left[\mathcal{O}\left(N \cdot D \cdot (k_{\max} \cdot \log N + k_{\max}/\delta + 1/\delta^3 + k_{\max}^2) + N \cdot C \cdot d^3 \right) \right].$$

Remember that $\hat{m}$ denotes the output estimate of SSV1, while $s_{\max}^{(2)} = 10$ is a constant parameter of SSV2. Moreover, $k_d = \max\{d + 3, 12\}$ and $k_{\max} = 30$ are the respective numbers of nearest neighbor points, while $C = 1000$ is the constant number of analyzed simplex volumes, and $\delta = 0.001$ is the fixed parameter gauging the precision of the

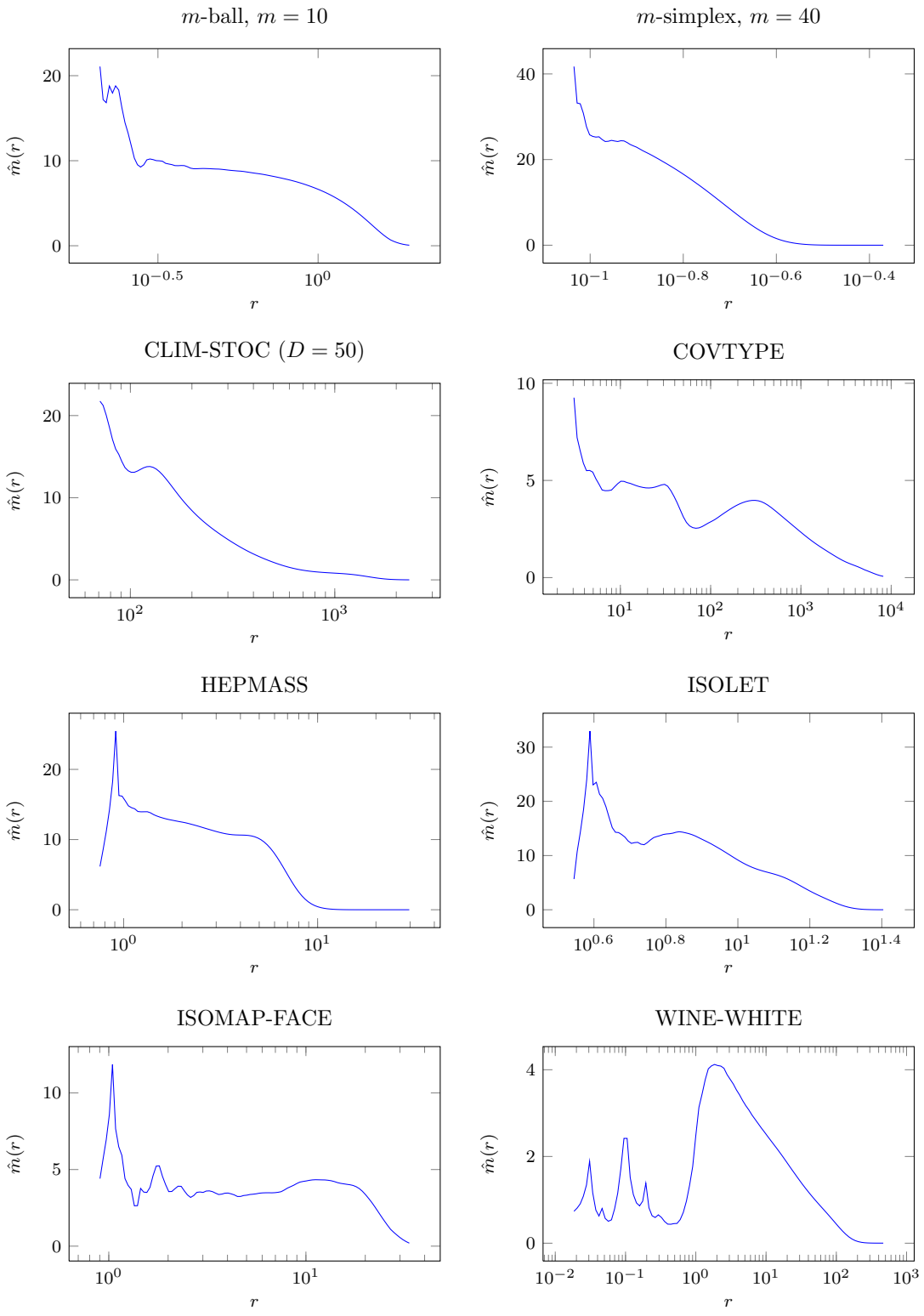

Figure 3.9: Scale-dependent correlation dimensions for selected datasets

bounding ball approximation.

Note further that the term $\mathcal{O}(N \cdot C \cdot d^3)$ is due to the evaluation of the simplex volumes and dominates the overall runtime of the SSV1 approach in case of input data with higher IDs m. As mentioned before, the simplex volume computation is naturally more expensive than the calculation of pairwise distances, which is why our SSV algorithms can generally not achieve a similar runtime performance as other methods completely relying on local distance computations, such as MLE, DD, and ESS.

When it comes to the actual runtime results, let us first emphasize that, for datasets of size $N > 1000$, both our algorithms SSV1 and SSV2 do not compute their pointwise estimates for *each* point, but rather for a reduced set of test points; this process has already been described in subsection 3.5.2 in full detail. Consequently, for a dataset of size $N = 10000$, roughly $10000/8 = 1250$ test points are considered.

All cpu time measurements have been performed on a machine with 16GB RAM and an Intel Xeon CPU E5-2620 v2 @2.10GHz featuring 6 cores, 12 threads and 15MB of L3-cache. No parallelization has been utilized in any form. The runtimes provided in table 3.24 include the entire algorithmic procedure, i.e., the tree construction and estimation process; the times required for reading the input data from harddrive and writing the output data are not comprised here. For each combination of parameters, we sampled 10 different datasets and we provide the average of the results of those 10 experiments. In order to analyze the main trends, we selected three candidates of the above described synthetic datasets: the m-ball as our reference object, the m-simplex as an evidentially hard to estimate structure, and the m-paraboloid because of its non-uniform sampling.

Let us now examine the empirical runtimes in table 3.24 of the first series of experiments, i.e., for fixed $D = 60$ and varying m. Here, while the SSV1 runtimes grow with increasing intrinsic dimensions $m = 4, 10, \ldots, 40$, the SSV2 runtimes remain nearly constant for increasing $m \geq 10$. Those characteristics are, of course, in full consensus with the theoretic cost complexities, in particular considering the outer sums capped by $\hat{m}$ and $s^{(2)}_{\max} = 10$, respectively. Furthermore, the differences in the outcomes between the three objects (ball, simplex, and paraboloid) are rather small. In fact, the differing runtimes of the SSV1 method (for higher values of m) are simply related with its distinct estimates, which are e.g. for $m = 40$: $\hat{m} = 40$ for the ball, $\hat{m} = 37$ for the simplex, and $\hat{m} = 38$ for the paraboloid.

The second series of experiments (fixed $m = 4$) shows another trend: while there is only a mild growth (at most about a factor of 2) in runtimes for increasing $D = 12, \ldots, 100$, we find a roughly linear time scaling with $D = 100, 1000, 10000$. Clearly, for lower ambient dimension D, the simplex volume computation with its complexity $\mathcal{O}(N \cdot C \cdot d^3)$, being independent of D, dominates the overall costs. In contrast, for $D \geq 100$, the costs for nearest neighbor search and bounding ball computations, which both scale linearly in D, are dominating. The speed-up of the SSV2 approach as compared to SSV1 for high

Table 3.24: CPU time measurements (in seconds) of SSV1 and SSV2 algorithms for synthetic datasets (m-ball, m-simplex, m-paraboloid embedded in $\mathbb{R}^D$) with fixed sampling number $N = 10000$ for varying intrinsic dimension m and ambient dimension D

		m-ball		m-simplex		m-paraboloid	
D	m	SSV1	SSV2	SSV1	SSV2	SSV1	SSV2
60	4	2.79	1.90	2.41	1.86	3.01	1.90
60	10	14.0	13.6	7.87	13.1	7.77	12.4
60	16	41.7	14.2	30.2	13.5	31.2	12.9
60	20	71.8	14.6	53.7	13.7	58.8	12.6
60	30	210	15.3	173	14.1	195	12.9
60	40	607	16.1	358	14.4	397	13.3
12	4	1.30	1.52	1.26	1.47	1.30	1.50
20	4	1.54	1.56	1.49	1.54	1.48	1.55
50	4	2.26	1.79	2.39	1.79	2.28	1.83
100	4	3.51	2.21	3.31	2.18	3.40	2.18
1000	4	24.2	9.61	23.7	9.52	24.2	9.48
10000	4	231	83.7	228	85.1	236	87.0
12	10	8.48	10.8	4.26	10.6	3.80	10.4
20	10	8.57	11.4	5.08	11.1	4.24	10.9
50	10	12.8	13.1	7.23	12.6	5.89	12.1
100	10	17.3	15.7	11.6	14.8	8.41	13.9
1000	10	95.2	65.1	82.3	55.8	53.6	51.0
10000	10	885	532	659	449	498	408

values of D can be traced back to the precise implementations.[3] Finally, the differences due to the three geometric objects are negligible in this scenario.

Regarding the third series (fixed $m = 10$), we observe the same correlation between runtimes and increasing D as before. Again, the SSV2 method is faster than SSV1 for higher ambient dimension D. Moreover, the slightly lower runtimes for the paraboloid data as compared to the other datasets are probably a consequence of a more efficient nearest neighbor search for the non-uniformly sampled points as opposed to the uniformly sampled ball and simplex.

As already mentioned in the descriptions of our SSV methods in subsection 3.4.5, the algorithmic design of the SSV1 approach has been optimized with regard to two aspects, which are accurate and reproducible estimation results, as well as a clear and easily comprehensible structure. The SSV2 approach generally yields a better runtime performance, which comes however at the expense of less accurate estimates for high IDs m, at least for the noisefree datasets considered here.

To conclude this short examination, let us propose three possibilities to considerably reduce the measured computation times, especially those of the SSV1 method. First, note that for the determination of the estimated output value $\hat{m}$ of some given dataset, ultimately, only the average d-simplex volumes for $d = \hat{m}$ and $d = \hat{m} + 1$ are relevant. Thus, if one expects a higher estimate $\hat{m}$, instead of evaluating all d-simplex volumes for $d = 1, \ldots, \hat{m} + 1$, one could rather use an adaptive scheme for the following three procedures: the selection of certain test dimensions d, the choice of (a fewer number of) test points, and a variable number C of sample simplices. This would be reasonable, since in case that the current test dimension d is "far away" from $\hat{m}$, only a small number of test points and sample simplices are required, as the empirical average volumes considerably deviate from the corresponding expected values.

A second attempt to improve the performance for higher values of the ambient dimension D could be made by using a more efficient nearest neighbor search structure. In our implementation, we opted for the *kd*-tree also for the sake of simplicity. Undoubtedly, a more advanced scheme would accelerate the entire estimation process without affecting the quality of the results.

Lastly, for very high ambient dimensions of $D = 1000$ and above, the computation of the approximate bounding balls becomes more and more expensive and could be replaced by a less precise approximation. For example, for the k NN points of some $\boldsymbol{x}_i$, one could consider the ball $B_r(\boldsymbol{x}_i)$ with center $\boldsymbol{x}_i$ and radius $r = T_k(\boldsymbol{x}_i)$, which is definitely a bounding ball, but usually not the *minimum* bounding ball for the $k + 1$ points. Obviously, this adjustment is likely to lead to slightly inferior estimation results. Nevertheless, a quantitative investigation of the effects could be interesting.

[3] Remember that the SSV2 algorithm computes all d-dimensional simplex volumes for $d = 1, \ldots, 5$ at once, meaning that the NN search must only be performed a single time. The SSV1 algorithm on the other hand starts a new NN search for each test dimension d.

Chapter 4

Application of the SSV Method in Dimensionality Reduction

In section 2.2 we introduced the concept of dimensionality reduction (DR) and outlined the importance of a reliable estimate of the intrinsic dimension (ID) for this purpose. After the presentation and thorough discussion of our new "Sample Simplex Volume" (SSV) approaches for intrinsic dimension estimation (IDE) in the preceding chapter, we now provide a final toy example to show the interplay between IDE and DR methods. Just like in Chapter 2, we choose a low-dimensional manifold which permits an improved visualization of the different dimensionality reduction results.

We consider a three-dimensional paraboloid embedded in $\mathbb{R}^4$. Multiple paraboloids of different dimensionalities have already been examined in our numerical experiments in subsection 3.5.4. As before, we apply a non-uniform sampling based on the exponential distribution. The exact sampling process is given by the following steps:

- Let $E_0, \ldots, E_3$ be *i.i.d.* according to the exponential distribution $\exp(\lambda)$ with rate parameter $\lambda = 1$;
- let $X_j := (1 + E_j/E_0)^{-1}$ for $j = 1, \ldots, 3$;
- let $X_4 := X_1^2 + (4X_2)^2 + (16X_3)^2$.

Note that this paraboloid is strongly distorted (or "stretched out") due to the definition of X_4. Our dataset consists of $N = 10000$ randomly sampled points. Figure 4.1 shows two different representations of the data. In the first plot, the first three coordinates x_1, x_2, x_3 are plotted to visualize the non-uniform sampling, while the last coordinate x_4 is represented by the coloring. Clearly, the value of x_4 mainly depends on the x_3 variable as expected. The second plot represents the coordinates x_1, x_3, x_4, where now, x_2 is represented by the coloring. Note here, that the x_4-axis features a completely different scale than the other two axes. The chosen perspective allows to see the curvature of the paraboloid.

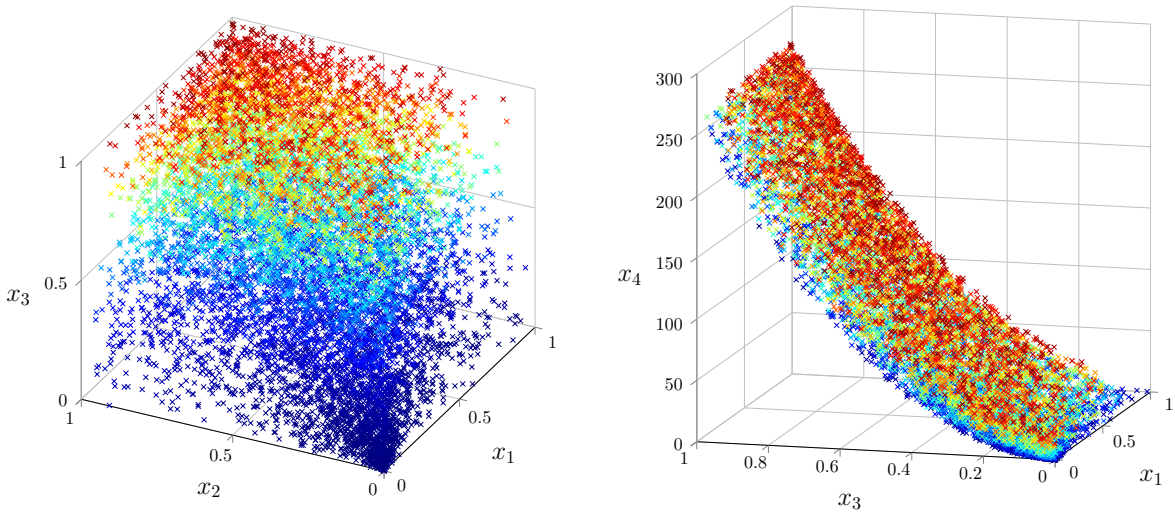

Figure 4.1: Three-dimensional representations of a paraboloid embedded in $\mathbb{R}^4$, sampled with 10000 points. *Left:* plot of coordinates (x_1, x_2, x_3), coloring symbolizes x_4. *Right:* plot of coordinates (x_1, x_3, x_4), coloring symbolizes x_2.

Table 4.1 shows the corresponding ID estimates of the methods introduced and compared above. The results are the empirical average of 10 different samplings of the same paraboloid, where each individual outcome has been rounded to its nearest integer (in case of a non-integer output value). Some methods underestimate the true number of latent variables (i.e., the underlying intrinsic dimension) as $\hat{m} = 2$ or even $\hat{m} = 1$ in case of the CD estimator. The reason can partly be traced back to the non-uniform sampling, but primarily to the highly unequal influence of the three variables X_1, X_2, X_3 onto X_4.

Finally, figure 4.2 shows plots of the three-dimensional embedding of the paraboloid, computed by classical multi-dimensional scaling (MDS) and the ISOMAP method (with NN parameter $k = 12$), respectively. Each embedding is visualized three times, where the coloring varies according to the three variables x_1, x_2, x_3. In fact, as can be seen from the clean color gradients, the MDS approach is able to produce a quite satisfying low-dimensional representation, where each of the three variables corresponds to a specific side of the cuboid, which the point set roughly resembles. The ISOMAP embedding

Table 4.1: IDE results for the three-dimensional paraboloid

D	m	CD	GCD	MLE	DD	ANC	ESS	SSV1	SSV2
4	3	1.6	2	3	3	3	2	3	3

on the other hand somehow reflects the curvature of the paraboloid, while — on the downside — both the x_1- and x_2-variables are considerably mixed up in the embedded point set. It is noteworthy that the result produced by the *linear* MDS method allows for a much better separation of the generating variables than that of the *nonlinear* ISOMAP approach, even though the underlying manifold is nonlinear.

However, the objective here is not the qualitative comparison between the outcomes of MDS and ISOMAP, but rather between the three-dimensional and two-dimensional embeddings achieved by those methods. The latter are plotted in figure 4.3. Obviously, since both MDS and ISOMAP calculate hierarchical solutions, the two-dimensional sets are mere projections of their three-dimensional counterparts. Yet, the crucial point is the fact that — for both selected DR approaches — the two-dimensional embeddings lead to a loss of information that has been present in the original dataset. While the 3D-embeddings feature the major spread due to the x_3-variable as well as the *two different* minor spreads due to the x_1- and x_2-variables, those two are inevitably mingled in the 2D-embeddings.

This final example again proves the significance of reliable intrinsic dimension estimators, such as our two SSV approaches introduced above, to guarantee the proper functioning of dimensionality reduction methods, which themselves continue to become more important in the field of high-dimensional data mining.

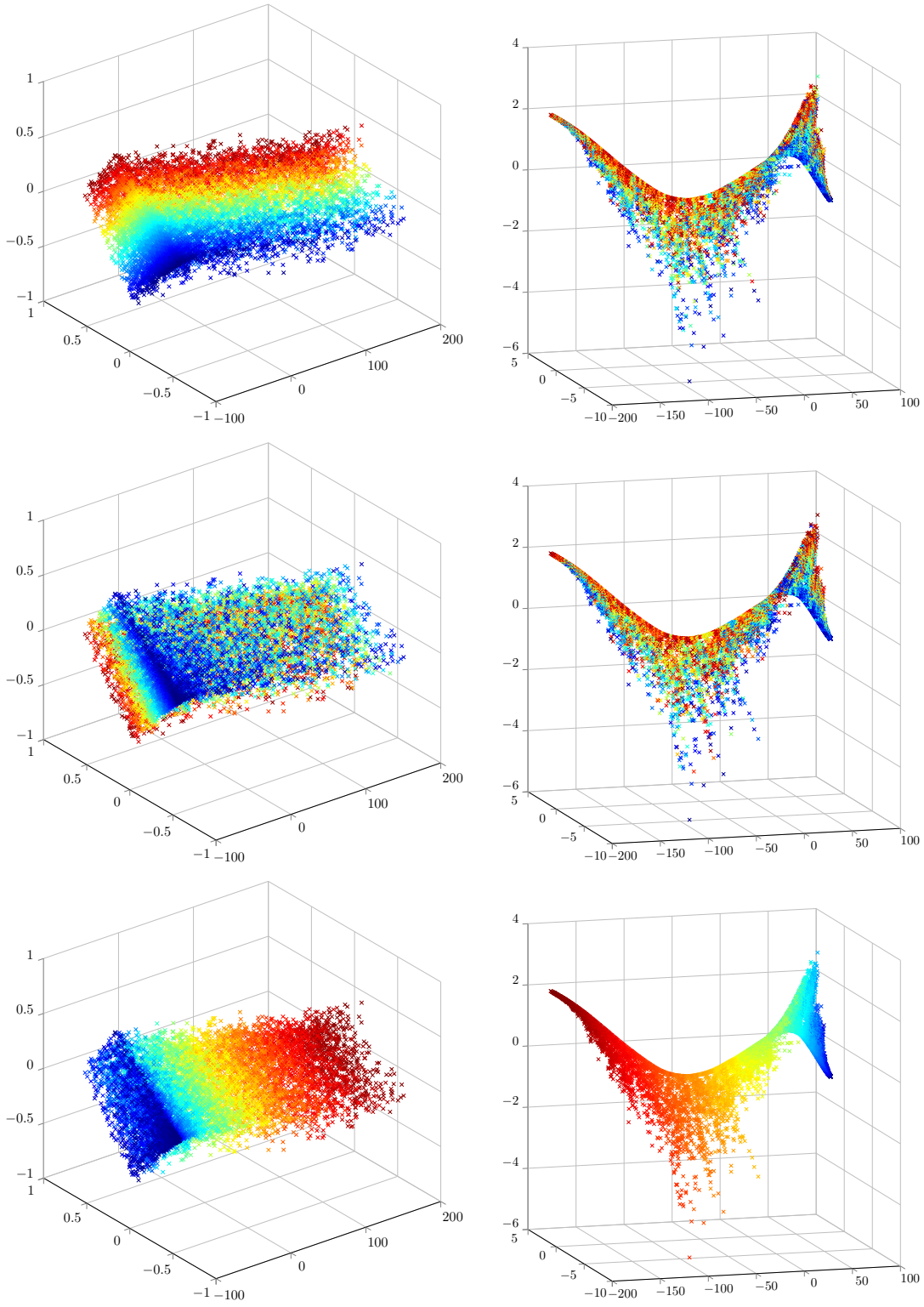

Figure 4.2: Three-dimensional embedding of the paraboloid computed by MDS *(left column)* and ISOMAP with $k = 12$ *(right column)*; coloring according to latent variables x_1 *(top)*, x_2 *(middle)*, and x_3 *(bottom)*, respectively.

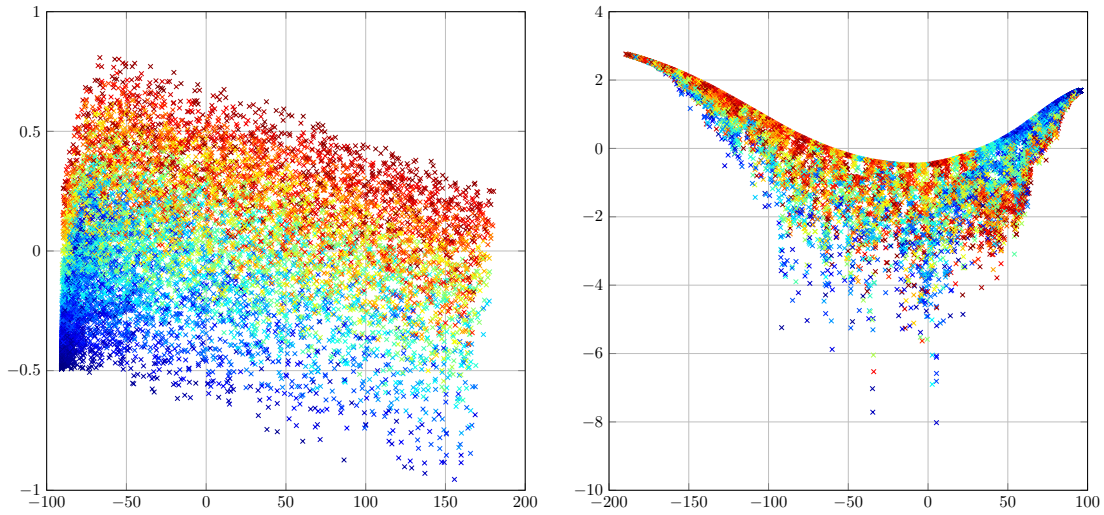

Figure 4.3: Two-dimensional embedding computed by MDS *(left)* and ISOMAP with $k = 12$ *(right)*; coloring according to latent variable x_1.

Chapter 5

Conclusion

Summary Let us now give a compact summary of our studies and especially the comprised main results. The focus of this work lies on the task of intrinsic dimension estimation and the development of a new approach for this purpose. Since the term *intrinsic dimension* of an arbitrary point set is not a precisely defined mathematical concept, we first recapitulated the most common and important notions of dimension. Among the multitude of existing dimension estimation techniques, we chose six particular ones, including a classical correlation dimension estimator, a widely used maximum likelihood method, and recent approaches based on angular distributions and simplex skewness, respectively. These methods were described in full detail and examined in the subsequent numerical experiments.

In the central part of our work, we introduced our new estimation approach where we first motivated the use of arbitrarily high-dimensional simplex volumes with randomly sampled vertex points. We derived an algorithm for the fast evaluation of multiple simplex volumes such that its overall workload is rather dominated by the *intrinsic* dimension, but not the *ambient* (original) dimension of the dataset. Based on this core component, we developed two algorithms, called *Sample Simplex Volume* methods, which allow to derive an estimate via comparing the average of random simplex volumes in local regions to the respective theoretically expected values. While the first variant is a straightforward procedure, the second one basically represents a well balanced trade-off between estimation precision and runtime performance.

In our comprehensive numerical examinations, we compared our new techniques against the six above-mentioned selected methods. We considered a large spectrum of low- and high-dimensional synthetical data, where both our approaches achieved excellent results, and especially the first one usually outperformed its competitors. When it comes to data tainted with Gaussian noise, the accuracy of all tested methods relying on the measurement of local quantities suffered to a similar degree. Naturally, without any a priori knowledge about the data, an increasing amount of high-dimensional noise renders a proper estimation virtually impossible. The results for real-world datasets confirmed

this insight; moreover, they verified the general competitiveness of our new estimators and their reliability for less noisy data.

Outlook Our approaches for intrinsic dimension estimation have demonstrated a reliable performance. They are based on straightforward geometric considerations and thus still leave space for further improvements and extensions. In order to decrease the expected runtime, one could employ an adaptive scheme to select optimized values for the hitherto constant parameters, i.e., the number of test simplices and the tolerance parameter, compare also the discussion at the end of subsection 3.4.5. Furthermore, the greedy incrementing of the test dimension in the SSV1 variant could be replaced by a more advanced search technique switching between low and high values, resulting in drastic time savings in the case of higher intrinsic dimensions.

Both our methods rely on the central theorem 3.7 by Miles, which specifies the expected volume of an arbitrary s-dimensional simplex with vertex points drawn at random from the uniform distribution over some D-dimensional ball. The most obstructive constraint here certainly is the assumption about *uniformly* distributed points. Consequently, a straightforward multiscale generalization of our approach is not reasonable, as this assumption is only valid in local regions for general datasets. Nevertheless, varying the local region size and comparing the respective empirical average volumes could provide additional knowledge about the dataset. One possible goal could be the estimation of the noise level and eventually the denoising of the data.

Moreover, a modification of our approach could be used to detect and separate low-dimensional affine subspaces or manifolds in high-dimensional data, sometimes referred to as *stratification learning* or *manifold clustering*.

Finally, our technique for the efficient computation of multiple simplex volumes might also turn out to be useful to accelerate the evaluation of the exact volume of high-dimensional convex polytopes, which are — for this purpose — usually represented as unions and intersections of many simplices, compare [BEF00]. Naturally, the improvement would only consist of a constant factor, which could still be significant in case of polytopes composed of a huge number of simplices.

Bibliography

[AHK01] Aggarwal, C.C., Hinneburg, A., and Keim, D.A.: *On the surprising behavior of distance metrics in high dimensional space.* In J. Van den Bussche and V. Vianu (editors), *Database Theory – ICDT 2001, 8th International Conference, Proceedings*, pp. 420–434. Springer Berlin Heidelberg, 2001

[AHM+11] Alwall, J., Herquet, M., Maltoni, F., Mattelaer, O., and Stelzer, T.: *MadGraph 5: going beyond.* Journal of High Energy Physics, 2011: 128, June 2011

[Aik91] Aikawa, H.: *Quasiadditivity of Riesz capacity.* Mathematica Scandinavica, 69: 15–30, 1991

[ARM97] Aguirre, L.A., Rodrigues, G.G., and Mendes, E.M.A.M.: *Nonlinear identification and cluster analysis of chaotic attractors from a real implementation of Chua's circuit.* International Journal of Bifurcation and Chaos, 7: 1411–1423, June 1997

[Ash99] Ashkenazy, Y.: *The use of generalized information dimension in measuring fractal dimension of time series.* Physica A: Statistical Mechanics and its Applications, 271: 427–447, September 1999

[Ass79] Assouad, P.: *Étude d'une dimension métrique liée à la possibilité de plongements dans* $\mathbb{R}^n$. Comptes Rendus de l'Académie des Sciences Paris Séries A, 288: 731–734, 1979

[Aud11] Audet, D.: *Déterminants sphérique et hyperbolique de Cayley-Menger.* Bulletin AMQ, LI, May 2011

[Bar68] Bareiss, E.H.: *Sylvester's identity and multistep integer-preserving Gaussian elimination.* Mathematics of Computation, 22, July 1968

[BBD99] Borovkova, S., Burton, R., and Dehling, H.: *Consistency of the Takens estimator for the correlation dimension.* The Annals of Applied Probability, 9: 376–390, May 1999

[BCF+16] Baldi, P., Cranmer, K., Faucett, T., Sadowski, P., and Whiteson, D.: *Parameterized neural networks for high-energy physics.* The European Physical Journal C, 76: 235, May 2016

[BD99] Blackard, J.A. and Dean, D.J.: *Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables.* Computers and Electronics in Agriculture, 24: 131–151, 1999

[BDJP79] Bailey, T.A., Dubes, R.C., Jain, A.K., and Pettis, K.W.: *An intrinsic dimensionality estimator from near-neighbor information.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 1: 25–37, January 1979

[BEF00] Büeler, B., Enge, A., and Fukuda, K.: *Exact volume computation for polytopes: A practical study.* In G. Kalai and G.M. Ziegler (editors), *Polytopes — Combinatorics and Computation*, pp. 131–154. Birkhäuser Basel, 2000

[Ben65] Bennett, R.S.: *Representation and analysis of signals – Part XXI: The intrinsic dimensionality of signal collections.* Tech. Rep. AD0475844, Department of Electrical Engineering and Computer Science, The Johns Hopkins University, Baltimore, December 1965

[Ben69] Bennett, R.S.: *The intrinsic dimensionality of signal collections.* IEEE Transactions on Information Theory, 15: 517–525, September 1969

[Ber09a] Berger, M.: *Geometry 1.* Springer-Verlag, 2009, 4th ed.

[Ber09b] Berger, M.: *Geometry 2.* Springer-Verlag, 2009, 4th ed.

[BG05] Borg, I. and Groenen, P.J.: *Modern Multidimensional Scaling: Theory and Applications.* Springer, New York, 2005, 2nd ed.

[BGG16] Bohn, B., Garcke, J., and Griebel, M.: *A sparse grid based method for generative dimensionality reduction of high-dimensional data.* Journal of Computational Physics, 309: 1–17, March 2016

[BGRS99] Beyer, K., Goldstein, J., Ramakrishnan, R., and Shaft, U.: *When is "nearest neighbor" meaningful?* In C. Beeri and P. Buneman (editors), *Database Theory – ICDT'99: 7th International Conference, Proceedings*, pp. 217–235. Springer Berlin Heidelberg, 1999

[BHH59] Beardwood, J., Halton, J., and Hammersley, J.: *The shortest path through many points.* Mathematical Proceedings of the Cambridge Philosophical Society, 55: 299–327, October 1959

[Bis99] Bishop, C.M.: *Bayesian PCA.* In M.J. Kearns, S.A. Solla, and D.A. Cohn (editors), *Advances in Neural Information Processing Systems 11*, pp. 382–388. MIT Press, 1999

[BL05] Bickel, P.J. and Levina, E.: *Maximum likelihood estimation of intrinsic dimension.* In L. Bottou, L.K. Saul, and Y. Weiss (editors), *Advances in Neural Information Processing Systems 17*, pp. 777–784. MIT Press, 2005

[BM15] Biau, G. and Mason, D.M.: *High-dimensional p-norms.* In M. Hallin, D.M. Mason, D. Pfeifer, and J.G. Steinebach (editors), *Mathematical Statistics and Limit Theorems: Festschrift in Honour of Paul Deheuvels*, pp. 21–40. Springer International Publishing, 2015

[BQY02] Brito, M.R., Quiroz, A.J., and Yukich, J.E.: *Graph-theoretic procedures for dimension identification.* Journal of Multivariate Analysis, 81: 67–84, April 2002

[BQY13] Brito, M.R., Quiroz, A.J., and Yukich, J.E.: *Intrinsic dimension identification via graph-theoretic methods.* Journal of Multivariate Analysis, 116: 263–277, April 2013

[BS98] Bruske, J. and Sommer, G.: *Intrinsic dimensionality estimation with optimally topology preserving maps.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 20: 572–575, May 1998

[CA74] Chen, C.K. and Andrews, H.C.: *Nonlinear intrinsic dimensionality computations.* IEEE Transactions on Computers, 23: 178–184, February 1974

[Cam03] Camastra, F.: *Data dimensionality estimation methods: a survey.* Pattern Recognition, 36: 2945–2954, December 2003

[CC00] Cox, T.F. and Cox, M.A.: *Multidimensional Scaling.* Chapman and Hall/CRC, 2000, 2nd ed.

[CC07] Choi, H. and Choi, S.: *Robust kernel Isomap.* Pattern Recognition, 40: 853–862, March 2007

[CC09] Cheng, S.W. and Chiu, M.K.: *Dimension detection via slivers.* In *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 1001–1010. 2009

[CCA+09] Cortez, P., Cerdeira, A., Almeida, F., Matos, T., and Reis, J.: *Modeling wine preferences by data mining from physicochemical properties.* Decision Support Systems, 47: 547–553, November 2009

[CCB+14] Campadelli, P., Ceruti, C., Bassis, S., Casiraghi, E., Lombardi, G., and Rozza, A.: *DANCo: an intrinsic dimensionality estimator exploiting angle and norm concentration.* Pattern Recognition, 47: 2569–2581, 2014

[CCC+11] Campadelli, P., Ceruti, C., Casiraghi, E., Lombardi, G., and Rozza, A.: *Minimum neighbor distance estimators of intrinsic dimension.* In D. Gunopulos, T. Hofmann, D. Malerba, and M. Vazirgiannis (editors), *European Conference on Machine learning and Knowledge Discovery in Databases, Proceedings, Part II*, pp. 374–389. Springer Berlin Heidelberg, 2011

[CCC+12] Campadelli, P., Ceruti, C., Casiraghi, E., Lombardi, G., and Rozza, A.: *Novel high intrinsic dimensionality estimators.* Machine Learning, 89: 37–65, October 2012

[CCCR15] Campadelli, P., Ceruti, C., Casiraghi, E., and Rozza, A.: *Intrinsic dimension estimation: Relevant techniques and a benchmark framework.* Mathematical Problems in Engineering, 2015: to appear, 2015

[CCL+11] Campadelli, P., Casiraghi, E., Lombardi, G., Rosa, M., and Rozza, A.: *IDEA: intrinsic dimension estimation algorithm.* In G. Maino and G.L. Foresti (editors), *16th International Conference on Image Analysis and Processing, Proceedings, Part I*, pp. 433–442. Springer Berlin Heidelberg, 2011

[CDE+00] Cheng, S.W., Dey, T.K., Edelsbrunner, H., Facello, M.A., and Teng, S.H.: *Sliver exudation.* Journal of the ACM, 47: 883–904, September 2000

[CDR05] Cheng, S.W., Dey, T.K., and Ramos, E.A.: *Manifold reconstruction from point samples.* In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 1018–1027. 2005

[CDS02] Collins, M., Dasgupta, S., and Schapire, R.E.S.: *A generalization of principal component analysis to the exponential family.* In T.G. Dietterich, S. Becker, and Z. Ghahramani (editors), *Advances in Neural Information Processing Systems 14*, pp. 617–624. MIT Press, 2002

[CF91] Cole, R. and Fanty, M.: *Spoken letter recognition.* In R.P. Lippmann, J.E. Moody, and D.S. Touretzky (editors), *Advances in Neural Information Processing Systems 3*, pp. 220–226. Morgan-Kaufmann, 1991

[CF09] Camastra, F. and Filippone, M.: *A comparative evaluation of nonlinear dynamics methods for time series prediction.* Neural Computing and Applications, 18: 1021–1029, November 2009

[CFF85] Cavendish, J.C., Field, D.A., and Frey, W.H.: *An approach to automatic three-dimensional finite element mesh generation.* International Journal for Numerical Methods in Engineering, 21: 329–347, February 1985

[CFG91] Croft, H.T., Falconer, K.J., and Guy, R.K.: *Unsolved Problems in Geometry.* Springer New York, 1991

[CFM90] Cole, R., Fanty, M., and Muthusamy, Y.: *The ISOLET spoken letter database.* Tech. Rep. CSE 90-004, Department of Computer Science and Engineering, Oregon Graduate Institute, March 1990

[CG15] Cunningham, J.P. and Ghahramani, Z.: *Linear dimensionality reduction: Survey, insights, and generalizations.* Journal of Machine Learning Research, 16: 2859–2900, January 2015

[CH04a] Costa, J.A. and Hero, A.O.: *Geodesic entropic graphs for dimension and entropy estimation in manifold learning.* IEEE Transactions on Signal Processing, 52: 2210–2221, August 2004

[CH04b] Costa, J.A. and Hero, A.O.: *Learning intrinsic dimension and intrinsic entropy of high-dimensional datasets.* In *Proceedings of the 12th European Signal Processing Conference*, pp. 369–372. IEEE, 2004

[CHG05] Costa, J.A., Hero, A.O., and Girotra, A.: *Estimating local intrinsic dimension with k-nearest neighbor graphs.* In *13th Workshop on Statistical Signal Processing*, pp. 417–422. IEEE, 2005

[CKM85] Chua, L., Komoru, M., and Matsumoto, T.: *The double scroll.* IEEE Transactions on Circuits and Systems, 32: 797–818, August 1985

[CKM86] Chua, L., Komoru, M., and Matsumoto, T.: *The double-scroll family.* IEEE Transactions on Circuits and Systems, 33: 1072–1118, November 1986

[CMS12] Cireşan, D., Meier, U., and Schmidhuber, J.: *Multi-column deep neural networks for image classification.* In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 3642–3649. IEEE, 2012

[CN00] Chávez, E. and Navarro, G.: *Measuring the dimensionality of general metric spaces.* Tech. Rep. TR/DCC-00-1, Department of Computer Science, University of Chile, 2000

[CNBYM01] Chávez, E., Navarro, G., Baeza-Yates, R., and Marroquín, J.L.: *Searching in metric spaces.* ACM Computing Surveys, 33: 273–321, September 2001

[COD99] Cattral, R., Oppacher, F., and Deugo, D.: *Rule acquisition with a genetic algorithm.* In *Proceedings of the 1999 Congress on Evolutionary Computation*, p. 129. IEEE, 1999

[COD02] Cattral, R., Oppacher, F., and Deugo, D.: *Evolutionary data mining with automatic rule generalization.* In *Recent Advances in Computers, Computing and Communications*, pp. 296–300. WSEAS Press, 2002

[CRH07] Carter, K.M., Raich, R., and Hero, A.O.: *De-biasing for intrinsic dimension estimation.* In *14th Workshop on Statistical Signal Processing*, pp. 601–605. IEEE, 2007

[CRH10] Carter, K.M., Raich, R., and Hero, A.O.: *On local intrinsic dimension estimation and its applications.* IEEE Transactions on Signal Processing, 58: 650–663, February 2010

[Cut93] Cutler, C.D.: *A review of the theory and estimation of fractal dimension.* In H. Tong (editor), *Dimension Estimation and Models*, pp. 1–107. World Scientific, 1993

[CV02] Camastra, F. and Vinciarelli, A.: *Estimating the intrinsic dimension of data with a fractal-based method.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 24: 1404–1407, October 2002

[CW90] Coppersmith, D. and Winograd, S.: *Matrix multiplication via arithmetic progressions.* Journal of Symbolic Computation, 9: 251–280, March 1990

[DAD04] Doherty, K., Adams, R., and Davey, N.: *Non-Euclidean norms and data normalisation.* In *Proceedings of the 12th European Symposium on Artificial Neural Networks*, pp. 181–186. 2004

[Dem94] Demartines, P.: *Analyse de Données par Réseaux de Neurones Auto-Organisés.* Ph.D. thesis, Institut National Polytechnique de Grenoble, 1994

[DGGZ03] Dey, T.K., Giesen, J., Goswami, S., and Zhao, W.: *Shape dimension and approximation from samples.* Discrete & Computational Geometry, 29: 419–434, February 2003

[DGH10] Das Gupta, M. and Huang, T.S.: *Regularized maximum likelihood for intrinsic dimension estimation.* In *Proceedings of the Twenty-Sixth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-10)*, pp. 220–227. AUAI Press, 2010

[DH97] Demartines, P. and Hérault, J.: *Curvilinear component analysis: a self-organizing neural network for nonlinear mapping of data sets.* IEEE Transactions on Neural Networks, 8: 148–154, January 1997

[DK09] Durrant, R.J. and Kabàn, A.: *When is 'nearest neighbour' meaningful: A converse theorem and implications.* Journal of Complexity, 25: 385–397, August 2009

[DM98] Dryden, I.L. and Mardia, K.V.: *Statistical Shape Analysis.* Wiley, 1998, 1st ed.

[Don00] Donoho, D.L.: *High-dimensional data analysis: The curses and blessings of dimensionality.* In *AMS Mathematical Challenges of the 21st Century*, pp. 1–33. 2000

[dST03] de Silva, V. and Tenenbaum, J.B.: *Global versus local methods in nonlinear dimensionality reduction.* In S. Becker, S. Thrun, and K. Obermayer (editors), *Advances in Neural Information Processing Systems 15*, pp. 721–728. MIT Press, 2003

[ECA16] ECA&D Project Team: *European Climate Assessment and Dataset.* http://eca.knmi.nl, 2016. [Online; accessed 2017-August-02]

[EN97] Emert, J. and Nelson, R.: *Volume and surface area for polyhedra and polytopes.* Mathematics Magazine, 70: 365–371, December 1997

[ER92] Eckmann, J.P. and Ruelle, D.: *Fundamental limitations for estimating dimensions and Lyapunov exponents in dynamical systems.* Physica D: Nonlinear Phenomena, 56: 185–187, May 1992

[Fal03] Falconer, K.: *Fractal Geometry - Mathematical Foundations and Applications.* John Wiley, 2003, 2nd ed.

[FCX12] France, S.L., Carroll, J.D., and Xiong, H.: *Distance metrics for high dimensional nearest neighborhood recovery: Compression and normalization.* Information Sciences, 184: 92–110, February 2012

[FGK03] Fischer, K., Gärtner, B., and Kutz, M.: *Fast smallest-enclosing-ball computation in high dimensions.* In G. Battista and U. Zwick (editors), *Algorithms - ESA 2003: 11th Annual European Symposium, Proceedings*, pp. 630–641. Springer Berlin Heidelberg, 2003

[FGK15] Fischer, K., Gärtner, B., and Kutz, M.: *Miniball (ESA 2003) C++ Library.* https://github.com/hbf/miniball, 2015. [Online; accessed 2017-August-02]

[Fis00] Fischer, G.: *Lineare Algebra*. Vieweg, 2000, 12th ed.

[FO71] Fukunaga, K. and Olsen, D.R.: *An algorithm for finding intrinsic dimensionality of data*. IEEE Transactions on Computers, C-20: 176–183, February 1971

[FOY83] Farmer, J.D., Ott, E., and Yorke, J.A.: *The dimension of chaotic attractors*. Physica D: Nonlinear Phenomena, 7: 153–180, May 1983

[FQZ09] Fan, M., Qiao, H., and Zhang, B.: *Intrinsic dimension estimation of manifolds by incising balls*. Pattern Recognition, 42: 780–787, May 2009

[FSA07] Farahmand, A.M., Szepesvári, C., and Audibert, J.Y.: *Manifold-adaptive dimension estimation*. In *Proceedings of the 24th International Conference on Machine Learning*, pp. 265–272. ACM, 2007

[FWV05] François, D., Wertz, V., and Verleysen, M.: *Non-Euclidean metrics for similarity search in noisy datasets*. In *Proceedings of the 13th European Symposium on Artificial Neural Networks*, pp. 339–344. 2005

[FWV07] François, D., Wertz, V., and Verleysen, M.: *The concentration of fractional distances*. IEEE Transactions on Knowledge and Data Engineering, 19: 873–886, July 2007

[Gär99] Gärtner, B.: *Fast and robust smallest enclosing balls*. In J. Nešetřil (editor), *Algorithms - ESA' 99: 7th Annual European Symposium, Proceedings*, pp. 325–338. Springer Berlin Heidelberg, 1999

[GB14] Groenen, P.J. and Borg, I.: *Past, present, and future of multidimensional scaling*. In J. Blasius and M. Greenacre (editors), *Visualization and Verbalization of Data*, pp. 95–116. Chapman and Hall/CRC, 2014

[GC16] Granata, D. and Carnevale, V.: *Accurate estimation of the intrinsic dimension using graph distances: Unraveling the geometric complexity of datasets*. Scientific Reports, 6: 1–12, August 2016

[GD09] Guan, Y. and Dy, J.G.: *Sparse probabilistic principal component analysis*. In *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics*, pp. 185–192. 2009

[Geo04] Georgii, H.O.: *Stochastik - Einführung in die Wahrscheinlichkeitstheorie und Statistik, 2. Auflage*. de Gruyter, 2004

[GKWZ08] Gorban, A.N., Kégl, B., Wunsch, D.C., and Zinovyev, A. (editors): *Principal Manifolds for Data Visualization and Dimension Reduction*. Lecture Notes in Computational Science and Engineering. Springer-Verlag, 2008

[GNU17] GNU: *GNU Scientific Library*. http://www.gnu.org/software/gsl/, 2017. [Online; accessed 2017-August-02]

[GP83] Grassberger, P. and Procaccia, I.: *Measuring the strangeness of strange attractors*. Physica D: Nonlinear Phenomena, 9: 189–208, October 1983

[Gro73] Groemer, H.: *On some mean values associated with a randomly selected simplex in a convex set*. Pacific Journal of Mathematics, 45: 525–533, 1973

[GVL96] Golub, G.H. and Van Loan, C.F.: *Matrix Computations*. Johns Hopkins University Press, 1996, 3rd ed.

[GW04] Giesen, J. and Wagner, U.: *Shape dimension and intrinsic metric from samples of manifolds*. Discrete & Computational Geometry, 32: 245–267, July 2004

[HA05] Hein, M. and Audibert, J.Y.: *Intrinsic dimensionality estimation of submanifolds in* $\mathbb{R}^d$. In *Proceedings of the 22nd International Conference on Machine Learning*, pp. 289–296. ACM, 2005

[HA16] Hein, M. and Audibert, J.Y.: *GCD intrinsic dimensionality estimation, C++ code*. http://www.ml.uni-saarland.de/code/IntDim/IntDim.htm, 2016. [Online; accessed 2017-August-02]

[HAK00] Hinneburg, A., Aggarwal, C.C., and Keim, D.A.: *What is the nearest neighbor in high dimensional spaces?* In *Proceedings of the 26th International Conference on Very Large Data Bases*, pp. 506–515. Morgan Kaufmann, 2000

[HC09] Hsu, C.M. and Chen, M.S.: *On the design and applicability of distance functions in high-dimensional data space*. IEEE Transactions on Knowledge and Data Engineering, 21: 523–536, April 2009

[HJ12] Horn, R.A. and Johnson, C.R.: *Matrix Analysis*. Cambridge University Press, 2012, 2nd ed.

[Hoe48] Hoeffding, W.: *A class of statistics with asymptotically normal distributions*. The Annals of Mathematical Statistics, 19: 293–325, September 1948

[HP83] Hentschel, H.G.E. and Procaccia, I.: *The infinite number of generalized dimensions of fractals and strange attractors*. Physica D: Nonlinear Phenomena, 8: 435–444, September 1983

[HR05] Howarth, P. and Rüger, S.: *Fractional distance measures for content-based image retrieval.* In D.E. Losada and J.M. Fernández-Luna (editors), *Advances in Information Retrieval: 27th European Conference on IR Research, Proceedings*, pp. 447–456. Springer Berlin Heidelberg, 2005

[HRS08] Haro, G., Randall, G., and Sapiro, G.: *Translated poisson mixture model for stratification learning.* International Journal of Computer Vision, 80: 358–374, December 2008

[Hug13] Hug, D.: *Random polytopes.* In E. Spodarev (editor), *Stochastic Geometry, Spatial Statistics and Random Fields: Asymptotic Methods*, chap. 7, pp. 205–238. Springer Berlin Heidelberg, 2013

[Jol02] Jolliffe, I.T.: *Principal Component Analysis.* Springer, New York, 2002, 2nd ed.

[JSF15] Johnsson, K., Soneson, C., and Fontes, M.: *Low bias local intrinsic dimension estimation from expected simplex skewness.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 37: 196–202, January 2015

[Kab12] Kabàn, A.: *Non-parametric detection of meaningless distances in high dimensional data.* Statistics and Computing, 22: 375–385, March 2012

[Kal92] Kaltofen, E.: *On computing determinants of matrices without divisions.* In *Proceedings of the International Symposium on Symbolic and Algebraic Computation*, pp. 342–349. 1992

[KB94] Koroljuk, V.S. and Borovskich, Y.V.: *Theory of U-Statistics.* Springer, 1994

[KD15] Karbauskaitė, R. and Dzemyda, G.: *Optimization of the maximum likelihood estimator for determining the intrinsic dimensionality of high-dimensional data.* International Journal of Applied Mathematics and Computer Science, 25: 895–913, December 2015

[KDM11] Karbauskaitė, R., Dzemyda, G., and Mazėtis, E.: *Geodesic distances in the maximum likelihood estimator of intrinsic dimensionality.* Nonlinear Analysis: Modelling and Control, 16: 387–402, December 2011

[Kég03] Kégl, B.: *Intrinsic dimension estimation using packing numbers.* In S. Becker, S. Thrun, and K. Obermayer (editors), *Advances in Neural Information Processing Systems 15*, pp. 697–704. MIT Press, 2003

[Kin69] Kingman, J.F.C.: *Random secants of a convex body.* Journal of Applied Probability, 6: 660–672, 1969

[KL51] Kullback, S. and Leibler, R.A.: *On information and sufficiency.* The Annals of Mathematical Statistics, 22: 79–86, March 1951

[Kle69] Klee, V.: *What is the expected volume of a simplex whose vertices are chosen at random from a given convex body?* The American Mathematical Monthly, 76: 286–288, 1969

[KLN+10] Kivimäki, I., Lagus, K., Nieminen, I.T., Väyrynen, J.J., and Honkela, T.: *Using correlation dimension for analysing text data.* In K. Diamantaras, W. Duch, and L.S. Iliadis (editors), *Artificial Neural Networks – ICANN 2010: 20th International Conference, Proceedings, Part I*, pp. 368–373. Springer Berlin Heidelberg, 2010

[Koh89] Kohonen, T.: *Self-Organization and Associative Memory.* Springer, Berlin Heidelberg, 1989, 3rd ed.

[Kru64a] Kruskal, J.B.: *Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis.* Psychometrika, 29: 1–27, March 1964

[Kru64b] Kruskal, J.B.: *Nonmetric multidimensional scaling: A numerical method.* Psychometrika, 29: 115–129, June 1964

[KS04] Kantz, H. and Schreiber, T.: *Nonlinear Time Series Analysis.* Cambridge University Press, 2004, 2nd ed.

[KTWK+02] Klein Tank, A.M.G., Wijngaard, J.B., Können, G.P., Böhm, R., Demarée, G., Gocheva, A., Mileta, M., Pashiardis, S., Hejkrlik, L., Kern-Hansen, C., Heino, R., Bessemoulin, P., Müller-Westermeier, G., Tzanakou, M., Szalai, S., Pálsdóttir, T., Fitzgerald, D., Rubin, S., Capaldo, M., Maugeri, M., Leitass, A., Bukantis, A., Aberfeld, R., van Engelen, A.F.V., Forland, E., Mietus, M., Coelho, F., Mares, C., Razuvaev, V., Nieplova, E., Cegnar, T., Antonio López, J., Dahlström, B., Moberg, A., Kirchhofer, W., Ceylan, A., Pachaliuk, O., Alexander, L.V., and Petrovic, P.: *Daily dataset of 20th-century surface air temperature and precipitation series for the European climate assessment.* International Journal of Climatology, 22: 1441–1453, 2002

[KV05] Kaltofen, E. and Villard, G.: *On the complexity of computing determinants.* Computational Complexity, 13: 91–130, February 2005

[KY79] Kaplan, J.L. and Yorke, J.A.: *Chaotic behavior of multidimensional difference equations.* In H.O. Peitgen and H.O. Walther (editors), *Functional Differential Equations and Approximation of Fixed Points*, vol. 730, pp. 204–227. Springer, Berlin, 1979

[LBBH98] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P.: *Gradient-based learning applied to document recognition.* Proceedings of the IEEE, 86: 2278–2324, November 1998

[Lee90] Lee, A.J.: *U-Statistics: Theory and Practice.* Marcel Dekker, Inc., 1990

[Lee97] Lee, J.M.: *Riemannian Manifolds: An Introduction to Curvature.* Springer, 1997

[LGX09] Li, C.G., Guo, J., and Xiao, B.: *Intrinsic dimensionality estimation within neighborhood convex hull.* International Journal of Pattern Recognition and Artificial Intelligence, 23: 31–44, February 2009

[Lic17] Lichman, M.: *UC Irvine Machine Learning Repository.* http://archive.ics.uci.edu/ml/, 2017. [Online; accessed 2017-August-02]

[LK13] Larsson, T. and Källberg, L.: *Fast and robust approximation of smallest enclosing balls in arbitrary dimensions.* In *Proceedings of the Eleventh Eurographics/ACMSIGGRAPH Symposium on Geometry Processing*, pp. 93–101. Eurographics Association, 2013

[LMR11] Little, A.V., Maggioni, M., and Rosasco, L.: *Multiscale geometric methods for estimating intrinsic dimension.* In *Proceedings SampTA*. 2011

[LMR12] Little, A.V., Maggioni, M., and Rosasco, L.: *Multiscale geometric methods for data sets I: Multiscale SVD, noise and curvature.* Journal of Machine Learning Research, 11: 411–450, September 2012

[Lom13] Lombardi, G.: *ANC intrinsic dimensionality estimation techniques, Matlab code.* http://www.mathworks.it/matlabcentral/fileexchange/40112, 2013. [Online; accessed 2017-August-02]

[LT13a] Lehrbäck, J. and Tuominen, H.: *A note on the dimensions of Assouad and Aikawa.* Journal of the Mathematical Society of Japan, 65: 343–356, 2013

[LT13b] Li, J. and Tao, D.: *Simple exponential family PCA.* IEEE Transactions on Neural Networks and Learning Systems, 24: 485–497, March 2013

[Luu98] Luukkainen, J.: *Assouad dimension: Antifractal metrization, porous sets, and homogeneous measures.* Journal of the Korean Mathematical Society, 35: 23–76, 1998

[LV07] Lee, J.A. and Verleysen, M.: *Nonlinear Dimensionality Reduction.* Springer-Verlag, 2007, 1st ed.

[LV10] Lee, J.A. and Verleysen, M.: *Unsupervised dimensionality reduction: Overview and recent advances.* In *International Joint Conference on Neural Networks*, pp. 1–8. IEEE, 2010

[LWC+07] Levina, E., Wagaman, A., Callender, A., Mandair, G., and Morris, M.: *Estimating the number of pure chemical components in a mixture by maximum likelihood.* Journal of Chemometrics, 21: 24–34, 2007

[Mat02] Matousek, J.: *Lectures on Discrete Geometry.* Springer-Verlag, 2002

[MF04] Michelucci, D. and Foufou, S.: *Using Cayley-Menger determinants for geometric constraint solving.* In *Proceedings of the Ninth ACM Symposium on Solid Modeling and Applications*, pp. 285–290. Eurographics Association, 2004

[MG05] MacKay, D.J. and Ghahramani, Z.: *Comments on 'Maximum likelihood estimation of intrinsic dimension' by E. Levina and P. Bickel*, 2005. http://www.inference.phy.cam.ac.uk/mackay/dimension/ [Online; accessed 2017-August-02]

[Mil71] Miles, R.E.: *Isotropic random simplices.* Advances in Applied Probability, 3: 353–382, 1971

[MM10] Mordohai, P. and Medioni, G.: *Dimensionality estimation, manifold learning and function approximation using tensor voting.* Journal of Machine Learning Research, 11: 411–450, March 2010

[MMN05] Mordohai, P., Medioni, G., and Nicolescu, M.: *The tensor voting framework.* In *Handbook of Geometric Computing*, pp. 535–568. Springer Berlin Heidelberg, 2005

[MN98] Matsumoto, M. and Nishimura, T.: *Mersenne twister: A 623-dimensionally equidistributed uniform pseudorandom number generator.* ACM Transactions on Modeling and Computer Simulation, 8(1): 3–30, 1998

[Mul59] Muller, M.E.: *A note on a method for generating points uniformly on n-dimensional spheres.* Communications of the ACM, 2: 19–20, April 1959

[Ott02] Ott, E.: *Chaos in Dynamical Systems.* Cambridge University Press, 2002, 2nd ed.

[OV13] Orsenigo, C. and Vercellis, C.: *A comparative study of nonlinear manifold learning methods for cancer microarray data classification.* Expert Systems with Applications, 40: 2189–2197, May 2013

[PCFS80] Packard, N., Crutchfield, J., Farmer, J., and Shaw, R.: *Geometry from a time series.* Physical Review Letters, 45: 712–716, 1980

[Pea90] Peano, G.: *Sur une courbe, qui remplit toute une aire plane.* Mathematische Annalen, 36: 157–160, 1890

[Pes93] Pesin, Y.B.: *On rigorous mathematical definitions of correlation dimension and generalized spectrum for dimensions.* Journal of Statistical Physics, 71: 529–547, May 1993

[Pes07] Pestov, V.: *Intrinsic dimension of a dataset: what properties does one expect?* In *Proceedings of the International Joint Conference on Neural Networks*, pp. 2959–2964. IEEE, 2007

[Pes08] Pestov, V.: *An axiomatic approach to intrinsic dimension of a dataset.* Neural Networks, 21: 204–213, March–April 2008

[Pfi89] Pfiefer, R.E.: *The historical development of J. J. Sylvester's four point problem.* Mathematics Magazine, 62: 309–317, 1989

[PM05] Papadopoulos, A.N. and Manolopoulos, Y.: *Nearest Neighbor Search: A Database Perspective.* Springer, 2005

[PP02] Polito, M. and Perona, P.: *Grouping and dimensionality reduction by locally linear embedding.* In T.G. Dietterich, S. Becker, and Z. Ghahramani (editors), *Advances in Neural Information Processing Systems 14*, pp. 1255–1262. MIT Press, 2002

[PY13] Penrose, M.D. and Yukich, J.E.: *Limit theory for point processes in manifolds.* The Annals of Applied Probability, 23: 2161–2211, December 2013

[PYS97] Pan, V.Y., Yu, Y., and Stewart, C.: *Algebraic and numerical techniques for the computation of matrix determinants.* Computers Math. Applic., 34, July 1997

[Rén59] Rényi, A.: *On the dimension and entropy of probability distributions.* Acta Mathematica Academiae Scientiarum Hungarica, 10: 193–215, 1959

[RL06] Raginsky, M. and Lazebnik, S.: *Estimation of intrinsic dimensionality using high-rate vector quantization.* In Y. Weiss, B. Schölkopf, and J.C. Platt (editors), *Advances in Neural Information Processing Systems 18*, pp. 1105–1112. MIT Press, 2006

[Rot01] Rote, G.: *Division-free algorithms for the determinant and the Pfaffian: Algebraic and combinatorial approaches.* In *Computational Discrete Mathematics*, pp. 119–135. Springer-Verlag, 2001

[RS00] Roweis, S.T. and Saul, L.K.: *Nonlinear dimensionality reduction by locally linear embedding.* Science, 290: 2323–2326, December 2000

[RS03] Roweis, S.T. and Saul, L.K.: *Think globally, fit locally: unsupervised learning of low dimensional manifolds.* Journal of Machine Learning Research, 4: 119–155, December 2003

[Rub82] Rubinstein, R.Y.: *Generating random vectors uniformly distributed inside and on the surface of different regions.* European Journal of Operational Research, 10: 205–209, June 1982

[Rud87] Rudin, W.: *Real and Complex Analysis.* McGraw-Hill, Inc., 1987, 3rd ed.

[Sam42] Samuelson, P.A.: *A method of determining explicitly the coefficients of the characteristic equation.* Ann. Math. Statist., 13, 1942

[Sam69] Sammon, J.W.: *A nonlinear mapping for data structure analysis.* IEEE Transactions on Computers, C-18: 401–409, May 1969

[SC66] Shepard, R.N. and Carroll, J.D.: *Parametric representation of nonlinear data structures.* In P.R. Krishnaiah (editor), *Multivariate Analysis*, pp. 561–592. Academic Press, New York, 1966

[Sch08] Schneider, R.: *Recent results on random polytopes.* Bollettino dell'Unione Matematica Italiana, 1: 17–40, 2008

[SDI06] Shakhnarovich, G., Darrell, T., and Indyk, P. (editors): *Nearest-Neighbor Methods in Learning and Vision: Theory and Practice.* MIT Press, 2006

[SG02] Stegmann, M.B. and Gomez, D.D.: *A brief introduction to statistical shape analysis.* Tech. rep., Informatics and Mathematical Modelling, University of Denmark, DTU, 2002

[SGM04] Saxena, A., Gupta, A., and Mukerjee, A.: *Non-linear dimensionality reduction by locally linear Isomaps.* In N.R. Pal, N. Kasabov, R.K. Mudi, S. Pal, and S.K. Parui (editors), *Neural Information Processing: 11th International Conference, Proceedings*, pp. 1038–1043. Springer Berlin Heidelberg, 2004

[Sma05] Small, M.: *Applied Nonlinear Time Series Analysis - Applications in Physics, Physiology and Finance.* World Scientific Publishing, 2005

[Smi88] Smith, L.A.: *Intrinsic limits on dimension calculations.* Physics Letters A, 133: 283–288, November 1988

[Söd11] Södergren, A.: *On the distribution of angles between the N shortest vectors in a random lattice.* Journal of the London Mathematical Society, 84: 749–764, 2011

[SRH10] Sricharan, K., Raich, R., and Hero, A.O.: *Optimized intrinsic dimension estimator using nearest neighbor graphs.* In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pp. 5418–5421. IEEE, 2010

[SSE87] Steele, M.J., Shepp, L.A., and Eddy, W.F.: *On the number of leaves of a Euclidean minimal spanning tree.* Journal of Applied Probability, 24: 809–826, 1987

[SSM98] Schölkopf, B., Smola, A., and Müller, K.R.: *Nonlinear component analysis as a kernel eigenvalue problem.* Neural Computation, 10: 1299–1319, July 1998

[SSV07] Sankaranarayanan, J., Samet, H., and Varshney, A.: *A fast all nearest neighbor algorithm for applications involving large point-clouds.* Computers and Graphics, 31: 157–174, April 2007

[Sto05] Stoer, J.: *Numerische Mathematik 1, 9. Auflage.* Springer-Verlag, 2005

[SVP14] Sorzano, C.O.S., Vargas, J., and Pascual?Montano, A.: *A survey of dimensionality reduction techniques*, 2014. Preprint arXiv:1403.2877

[SWLH10] Shen, H., Wang, L., Liu, Y., and Hu, D.: *Discriminative analysis of resting-state functional connectivity patterns of schizophrenia using low dimensional embedding of fMRI.* NeuroImage, 49: 3110–3121, February 2010

[Tak81] Takens, F.: *Detecting strange attractors in turbulence.* In D. Rand and L.S. Young (editors), *Dynamical Systems and Turbulence*, vol. 898 of *Lecture Notes in Mathematics*, pp. 366–381. Springer, 1981

[Tak85] Takens, F.: *On the numerical determination of the dimension of an attractor.* In B.L.J. Braaksma, H.W. Broer, and F. Takens (editors), *Dynamical Systems and Bifurcations*, vol. 1125 of *Lecture Notes in Mathematics*, pp. 99–106. Springer, 1985

[TB99] Tipping, M.E. and Bishop, C.M.: *Probabilistic principal component analysis.* Journal of the Royal Statistical Society. Series B (Statistical Methodology), 61: 611–622, 1999

[TdSL00] Tenenbaum, J.B., de Silva, V., and Langford, J.C.: *A global geometric framework for nonlinear dimensionality reduction.* Science, 290: 2319–2323, December 2000

[Ten98] Tenenbaum, J.B.: *Mapping a manifold of perceptual observations.* In *Advances in Neural Information Processing Systems (NIPS 1997)*, vol. 10, pp. 682–688. MIT Press, 1998

[TFV15] Tóth, G.F., Fodor, F., and Vígh, V.: *The packing density of the n-dimensional cross-polytope.* Discrete & Computational Geometry, 54: 182–194, July 2015

[The90] Theiler, J.: *Statistical precision of dimension estimators.* Physical Review A, 41: 3038–3051, March 1990

[TMGM06] Tatti, N., Mielikäinen, T., Gionis, A., and Mannila, H.: *What is the dimension of your binary data?* In *Sixth International Conference on Data Mining*, pp. 603–612. IEEE, 2006

[Tor52] Torgerson, W.S.: *Multidimensional scaling: I. Theory and method.* Psychometrika, 17: 401–419, December 1952

[Tru68] Trunk, G.V.: *Statistical estimation of the intrinsic dimensionality of data collections.* Information and Control, 12: 508–525, May 1968

[Tru76] Trunk, G.V.: *Statistical estimation of the intrinsic dimensionality of a noisy signal collection.* IEEE Transactions on Computers, C-25: 165–171, February 1976

[Vai89] Vaidya, P.M.: *An $O(n \log n)$ algorithm for the all-nearest-neighbors problem.* Discrete and Computational Geometry, 4: 101–115, March 1989

[VD95] Verveer, P.J. and Duin, R.P.W.: *An evaluation of intrinsic dimensionality estimators.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 17: 81–86, January 1995

[vdMPvdH09] van der Maaten, L.J., Postma, E.O., and van den Herik, H.J.: *Dimensionality reduction: A comparative review.* Tech. Rep. TiCC-TR 2009-005, Tilburg Center for Cognition and Communication, Tilburg University, October 2009

[Ver03] Verleysen, M.: *Learning high-dimensional data.* In S. Ablameyko, M. Gori, L. Goras, and V. Piuri (editors), *Limitations and Future Trends in Neural Computation*, pp. 141–162. IOS Press, 2003

[VKD09] Verma, N., Kpotufe, S., and Dasgupta, S.: *Which spatial partition trees are adaptive to intrinsic dimension?* In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, UAI '09, pp. 565–574. AUAI Press, 2009

[Wel91] Welzl, E.: *Smallest enclosing disks (balls and ellipsoids).* In H. Maurer (editor), *New Results and New Trends in Computer Science, Proceedings*, pp. 359–370. Springer Berlin Heidelberg, 1991

[WG94] Weigend, A.S. and Gershenfeld, N.A.: *The future of time series: Learning and understanding.* In A.S. Weigend and N.A. Gershenfeld (editors), *Time Series Prediction: Forecasting the Future and Understanding the Past*, pp. 1–70. Addison-Wesley, 1994

[Yil08] Yildirim, E.A.: *Two algorithms for the minimum enclosing ball problem.* SIAM Journal on Optimization, 19: 1368–1391, 2008

[ZADB06] Zezula, P., Amato, G., Dohnal, V., and Batko, M.: *Similarity Search: The Metric Space Approach*, vol. 32 of *Advances in Database Systems.* Springer, 2006

[ZHT06] Zou, H., Hastie, T., and Tibshirani, R.: *Sparse principal component analysis.* Journal of Computational and Graphical Statistics, 15: 265–286, 2006