Methods for Estimation of Intrinsic Dimensionality

KALANTAN, ZAKIAH,IBRAHIM

Methods for Estimation of Intrinsic Dimensionality

Zakiah Ebrahim Kalantan

A thesis presented for the degree of
Doctor of Philosophy

Statistics Group
Department of Mathematical Sciences
University of Durham
England

February 2014

Dedicated to

To my great loving parents..

To my dear husband Abdulmajeed..

To my dear and lovely kids (Nizar, Nael, Ryan, Anmar and Renad)..

To my lovely sisters and brother..

Methods for Estimation of Intrinsic Dimensionality

Zakiah Ebrahim Kalantan

Submitted for the degree of Doctor of Philosophy
February 2014

Abstract

Dimension reduction is an important tool used to describe the structure of complex data (explicitly or implicitly) through a small but sufficient number of variables, and thereby make data analysis more efficient. It is also useful for visualization purposes. Dimension reduction helps statisticians to overcome the ‘curse of dimensionality’. However, most dimension reduction techniques require the intrinsic dimension of the low-dimensional subspace to be fixed in advance.

The availability of reliable intrinsic dimension (ID) estimation techniques is of major importance. The main goal of this thesis is to develop algorithms for determining the intrinsic dimensions of recorded data sets in a nonlinear context. Whilst this is a well-researched topic for linear planes, based mainly on principal components analysis, relatively little attention has been paid to ways of estimating this number for non–linear variable interrelationships. The proposed algorithms here are based on existing concepts that can be categorized into *local methods*, relying on randomly selected subsets of a recorded variable set, and *global methods*, utilizing the entire data set.

This thesis provides an overview of ID estimation techniques, with special consideration given to recent developments in non–linear techniques, such as charting manifold and fractal–based methods. Despite their nominal existence, the practical implementation of these techniques is far from straightforward.

The intrinsic dimension is estimated via Brand’s algorithm by examining the growth point process, which counts the number of points in hyper-spheres. The estimation needs to determine the starting point for each hyper-sphere. In this thesis we provide settings for selecting starting points which work well for most data sets. Additionally we propose approaches for estimating dimensionality via Brand's algorithm, the Dip method and the Regression method.

Other approaches are proposed for estimating the intrinsic dimension by fractal dimension estimation methods, which exploit the intrinsic geometry of a data set. The most popular concept from this family of methods is the correlation dimension, which requires the estimation of the correlation integral for a ball of radius tending to 0. In this thesis we propose new approaches to approximate the correlation integral in this limit. The new approaches are the Intercept method, the Slop method and the Polynomial method.

In addition we propose a new approach, a localized global method, which could be defined as a local version of global ID methods. The objective of the localized global approach is to improve the algorithm based on a local ID method, which could significantly reduce the negative bias.

Experimental results on real world and simulated data are used to demonstrate the algorithms and compare them to other methodology. A simulation study which verifies the effectiveness of the proposed methods is also provided. Finally, these algorithms are contrasted using a recorded data set from an industrial melter process.

Declaration

The work in this thesis is based on research carried out at the Numerical Analysis Group, the Department of Mathematical Sciences, the Statistics and Probability group, England. No part of this thesis has been submitted elsewhere for any other degree or qualification and it all my own work unless referenced to the contrary in the text.

Acknowledgements

First, all praise and thanks to Allah my God, the Almighty, for blessing and giving me the strength to finish this work.

I would like to express my immense appreciation and gratitude to my supervisor Dr. Jochen Einbeck, for his guidance and great patience throughout my thesis. Dr. Einbeck has also offered considerable support throughout the process of publishing papers, and helped me during the revision of this thesis.

I would also like to express my gratitude to my second supervisor Professor Frank Coolen for his support and kindness. Special thanks to my local supervisor Dr. Samia Adham, King Abdulaziz University (KAU), for her guidance and help.

I also would like to thank Dr. Uwe Kruger, for supplying the literature that helped me to illustrate some of the methods used in the thesis, and for providing the data from an industrial glass melter process.

I am indebted and deeply grateful from the bottom of my heart to my mother, for her unlimited support, unconditional love, care for my childern and loving prayers for me during my education. My special thanks to my father, husband, sons and sisters (especially Maimunah and Izdihar) for their love, patience and unconditional support.

I would also like to thank all the staff of the Joint Supervision Program at KAU for their support and Durham university for providing the facilities that have helped me during my studies. Special thanks to my friend Tahani Coolen-Maturi for her help.

Finally, Many thanks go to my friend Sulafah Binhimd for her invaluable help and support throughout the time we spent together in Saudi Arabia and Durham during our doctoral research.

Zakiah Kalantan, Durham, UK.

Submission: September 2013.

Viva: November 2013.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Nowadays estimating intrinsic dimension plays an important role in many statistical applications such as pattern recognition or data mining algorithms. In this chapter we investigate the importance of the intrinsic dimension methods and provide an overview of the thesis.

1.1 Background and History

A real data set has to deal with very high-dimensional data which contains a large number of variables. In order to handle this data properly, we need to investigate whether or not it can be represented in a low-dimensional space. This step is very important since it alleviates the curse of dimensionality [4] and other issues such as increased computing time and data storage space.

The curse of dimensionality implies that several issues will arise when analyzing and visualizing data sets in high-dimensional spaces that do not occur in low-dimensional settings. The problems of high dimension are important in many fields such as data mining and machine learning. The common theme of those problems is that, when increasing number of variables one needs to adjust the sample size which is necessary for the data analysis. Those issues prevent efficient data analysis and organization. The technique to inhibit the curse of the dimensionality is to minimize the input dimension of the function to be estimated, using a small number of variables. Therefore, dimension reduction helps overcome the curse of dimensionality. One can observe that a particular variable, which is a part of a larger set, may contain information that is encapsulated in other variables too.

Dimension reduction models try to capture the significant information that is embedded within the recorded data set. Dimension reduction is often applied as a data pre–processing step or as a part of data analysis, capturing significant information in the original data, and then supporting the creation of reduced dimension data in the system. The main objective of dimension reduction is to transform the data space from a high-dimensional variable space into a low-dimensional space, so that the fundamental structure is easy to realize.

In 1901 Karl Pearson illustrated a technique to approximate data sets with straight lines and planes. He proposed a Principal Component Analysis (PCA) method, which is a fundamental of dimension reduction methods. Recently, several literature and further development methods have been proposed to obtain reduced dimension. Dimension reduction methods can be categorized as *linear* or *nonlinear* methods. The first type *linear methods* try to search globally flat subspace such as PCA. In the past few decades, various methods have been proposed for the linear data structure and these are mostly related to the application of PCA with several assumptions. The second type *nonlinear methods* try to search a locally flat subspace, such as multidimensional scaling methods and ISOMAP. Such methods require fixing the intrinsic dimension of the low-dimensional subspace in advance. As illustration of this is when the data points lie on a smooth curve, one can state that the intrinsic dimension equals 1 and that this is independent of the dimensionality of data representation.

The intrinsic dimension (ID) of a data set $Z \in \mathbb{R}^D$ can be defined as the minimum number of variables (d) necessary to describe the data without too much loss of information [8] [32]. Historically the ID used to be defined as equal to d when the data points lie entirely within an d-dimensional linear subspace of $\mathbb{R}^D$ [8], which is used to obtain ID for *linear* methods, for instances PCA, Factor analysis and Independent component analysis. We have in this thesis a more general notion in mind which comprises linear as well as nonlinear manifolds.

Intrinsic dimension methods try to eliminate the problem of high dimension.

Their advantages [8] are:

- a reduction in the size of the data storage space needed,
- faster computation because of fewer variables,
- the use of vectors with smaller dimensions often leads to improvements in the performance if further statistical inference, such as regression or classification, is to be carried out.

ID estimation methods can be classified into two groups; *local methods* which divide the data into small sub-regions, or provide a series of local ID estimates at several target points, in order to arrive at a suitably averaged overall ID estimator. Examples for such methods include Levina–Bickel's Maximum Likelihood estimator [60], Brand's concept of 'charting' [6], among others [32] [72], which propose concepts for estimating ID for subsets of a recorded data set.

In addition to *local methods*, a survey by Camastra [8] also emphasized that *global methods* can be considered. *Global methods* try to estimate the dimension using the whole data set, imposing the implicit assumption that the intrinsic dimension is constant over the data set. The methods can be further categorized into projection techniques, multidimensional scaling and fractal-based methods. It is interesting to note that $d \in \mathbb{R}$ in a nonlinear context. This family includes purely linear methods based on linear approximation (such as the 'broken stick method' and many other stopping rules for principal component analysis [46] [56]), and also non–parametric approaches such as fractal–based methods. The term 'fractal' is used since under this sort of approach, the intrinsic dimensionality d does not need to be an integer. Camastra presented a useful survey on intrinsic dimension estimation methods focusing on fractal-based methods [8] [9].

The most common route to fractal dimension estimation is via correlation dimension. The method requires the construction of a so–called correlation integral, from which the ID is extracted using appropriate techniques.

Although *nonlinear* methods (*global* or *local* methods) are available, it seems that not enough work has been devoted to practical implementations of the methodology of dimensionality estimation on non-linear manifolds. Furthermore, as with many methods, there is not enough evidence that they work well practically. Such as charting manifolds needs to satisfy target points. Additionally fractal methods require the construction of a correlation integral, from which the ID is extracted using appropriate techniques. This step is not straightforward, since it requires counting the number of data pairs within a ball of radius tending to 0.

1.2 Data

This section introduces some concepts in our approach to estimating intrinsic dimensionality. Let $X = (X_1, \ldots, X_D)^T$ be a random vector with mean and variance of X denoted by m and Σ, respectively. The random vector X has a probability density $g(x)$. Now a sample of size N is drawn from the random vector X, yielding data $Z = \{x_1, \ldots, x_N\} \in \mathbb{R}^D$, which are N independent and identically distributed (iid) observations. The matrix Z has the following structure:

$$Z = \begin{bmatrix} x_{11} & x_{12} & x_{13} & \ldots & x_{1j} & \ldots & x_{1D} \\ x_{21} & x_{22} & x_{23} & \ldots & x_{2j} & \ldots & x_{2D} \\ x_{31} & x_{32} & x_{33} & \ldots & x_{3j} & \ldots & x_{3D} \\ \vdots & \vdots & \vdots & \ldots & \vdots & \ldots & \vdots \\ x_{N1} & x_{N2} & x_{N3} & \ldots & x_{Nj} & \ldots & x_{ND} \end{bmatrix}$$

and could be written as

$$Z = \begin{pmatrix} x_1^T \\ \vdots \\ x_N^T \end{pmatrix}$$

where D is the number of variables and N is the number of observations. The mean of Z is denoted by

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i = \frac{1}{N}\sum_{i=1}^{N} \begin{pmatrix} x_{i1} \\ \vdots \\ x_{iD} \end{pmatrix} = \begin{pmatrix} \bar{x}_1 \\ \vdots \\ \bar{x}_D \end{pmatrix},$$

which is unbiased estimate of m. The maximum likelihood estimator of Σ is given by

$$\hat{\Sigma}_{ML} = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})(x_i - \bar{x})^T,$$

while the sample variance matrix is given by

$$\hat{\Sigma}_{sample} = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})(x_i - \bar{x})^T = \frac{N}{N-1}\hat{\Sigma}_{ML},$$

which is generally used. In practice, for this real data Z, Σ needs to be replaced by a suitable estimator $\hat{\Sigma}$. In this thesis, we use the notation Σ whether or not Σ was estimated.

1.3 General concepts

In this section the general definitions are briefly covered. In Subsection 1.3.1 the concepts of supervised and unsupervised learning are explained. The Subsection 1.3.2 presents the general definition of the probability density function. The Subsection 1.3.3 defines the multivariate density. Kernel density estimation is illustrated in the Subsection 1.3.4. Linear regression is briefly presented in the Subsection 1.3.5. The Subsection 1.3.6 illustrates the definition of polynomial regression model.

1.3.1 Supervised and unsupervised learning

There are two general concepts that are commonly used in machine learning; supervised and unsupervised learning algorithms. Supervised learning algorithms suppose that the observation is given in a training set of (input, output). Then the objective is to determine the function of output for invisible input patterns, which is a way of using concepts from Pearson's linear regression [36].

In contrast in unsupervised learning, there is only a set of input observations without a desired target [36]. Then one attempts to seek a good representation of the data, such as a reduction in the number of variables. It is noted that unsupervised learning can be much more challenging to manage than supervised learning [36]. The researcher in unsupervised learning usually faces a differentiation between representing the data as closely as possible and summarizing it as far as possible [36]. Manifold learning is unsupervised learning where the objective is to project the data into a new space (representation) which has a smaller dimension than the input space [36].

1.3.2 The probability density function

The probability distribution of a continuous random variable X is denoted as $g(x)$ and defined as

$$P\left(a \leq X \leq b\right) = \int_{b}^{a} g(x)dx.$$

Then $g(x)$ can be determined from a sample of data observations. This is done by using parametric approach or non-parametric approach.

The parametric approach means estimating $g(x)$ by assuming that X has probability distribution of one of a parametric distribution family. For instance one assumes that X has a normal distribution with parameters μ and σ, then the parameters are estimated from data set Z. Usually, this approach obtains steady estimates and is commonly used because it is easy to apply. The parametric approach has advantages as long as the assumption of the distribution is valid. Each parametric distribution requires some restrictions on the shape of $g(x)$, for instance with normal distribution, where the density curve $g(x)$ should be symmetric and bell-shaped. This disadvantage leads the researcher to propose non-parametric approaches.

Non-parametric approaches try to estimate $g(x)$ immediately from the data. The family of this approach includes histogram and kernel density estimation. The histogram is a commonly used and simple method. We will illustrate the kernel density estimation in the Subsection 1.3.4.

1.3.3 Multivariate probability density

As earlier we assume a random variable X, Section 1.2, which forms a $D-$dimensional random vector and

$$x = \begin{pmatrix} x_1 \\ \vdots \\ x_D \end{pmatrix}$$

is a particular realization of X. The probabilistic behaviour of X is entirely determined by the distribution function of X,

$$G(x) = G(x_1, \cdots, x_D) = P(X_1 \leq x_1, \cdots, X_D \leq x_D).$$

For a continuous random variable X, then there exists a probability density function $g : \mathbb{R}^D \to \mathbb{R}$ [21], such that

$$G(x) = \int_{-\infty}^{x} g(u)\, du \ = \int_{-\infty}^{x_D} \cdots \int_{-\infty}^{x_1} g(u_1, \cdots, u_D)\, du_1 \cdots du_D.$$

Then, for any subset $S \subset \mathbb{R}^D$ [21] one has

$$P(X \in S) = \int_S g(x)\, dx.$$

In particular, for $S = \mathbb{R}^D$,

$$\int_{\mathbb{R}^D} g(x)\, dx = 1.$$

1.3.4 Kernel density estimation

A univariate kernel density estimator for a random sample Z, defined in Section 1.2, drawn from X of the function $g(x)$ is defined as

$$\hat{g}(x; h) = \frac{1}{N} \sum_{i=1}^{N} K_h\left(x - x_i\right) = \frac{1}{Nh} \sum_{i=1}^{N} K\left(\frac{x - x_i}{h}\right),$$

where $K(\cdot)$ is the kernel function, which determines the shape of the weighting function. The parameter h is the fixed bandwidth which is a positive and non-random number. It determines the width of the weighting function and the amount of smoothing in estimating $g(x)$ [20]. Table 1.1 displays some of kernel functions.

Kernel	$K(x)$
Uniform	$\frac{1}{2}$ for $\lvert x\rvert < 1$, 0 otherwise
Triangle	$\frac{3}{4}(1-\lvert x\rvert)$ for $\lvert x\rvert < 1$, 0 otherwise
Epanechnikov	$\frac{3}{4}(1-x^2)$ for $\lvert x\rvert < 1$, 0 otherwise
Gaussian	$\frac{1}{\sqrt{2\pi}}\exp\left(-\frac{1}{2}x^2\right)$

Table 1.1: Some of Kernel functions.

In general the kernel functions are symmetric around 0 and integrate to 1 [37], and the bandwidth h should be small in order to reduce the bias of estimation.

Hence, the $D-$dimensional multivariate kernel density estimator [20] for a random sample $x_1, \cdots, x_N$ drawn from X, is

$$\hat{g}(x;\mathbf{H}) = \frac{1}{N}\sum_{i=1}^{N} K_{\mathbf{H}}\left(x - x_i\right),$$

where $x = (x_1, \cdots, x_D)^T$ and $x_i = (x_{i1}, \cdots, x_{iD})^T$, $i = 1, 2, \cdots, N$, while $K_{\mathbf{H}}(x)$ is defined as

$$K_{\mathbf{H}}(x) = \lvert\mathbf{H}\rvert^{-1/2} K\left(\mathbf{H}^{-1/2}\, x\right).$$

and $\mathbf{H}$ can be set equal to $\mathbf{H} = \text{diag}(h^2)$ if an equal degree of smoothing in all directions is desired. Here $K_{\mathbf{H}}$ is the scaled kernel and $\mathbf{H}$ is a $D \times D$ fixed bandwidth matrix which is a symmetric and a positive number [20].

1.3.5 Linear regression

Linear regression is a statistical method used to study the linear relationship between variables by fitting linear equations to the data points, based on the assumption that the errors of linear models are normally distributed. The linear equation has the form

$$y = b_0 + \ b_1 x + \ e,$$

where y is a scalar dependent variable and x is an explanatory variable. The parameters of the model are b_0 and b_1, where b_0 is the intercept and b_1 is the slope of the line.

Commonly, linear regression is fitted via the least squares method by minimizing the sum of squares of the vertical deviation from each data point to the line. The vertical deviations equals 0 when the data point lies on the fitted line. To display the fitted model, the computed regression line is plotted over data points. Once a regression model has been fitted then it could, with some caution, be used for extrapolation, which means predicting values that are outside the range of data set.

1.3.6 Polynomial regression model

The polynomial regression model is regarded as a special case of the multiple regression model when one independent variable is assumed. It can be considered as a Taylor series expansion of the unknown function. The model could be used as the approximation function of a complex nonlinear relationship. The polynomial regression of order p takes the form

$$y = b_0 + \ b_1x \ + \ b_2x^2 \ + \ b_3x^3 + \ \cdots + + b_px^p + \ e.$$

To decide the suitable value of p one can use the 'Multiple R^2' or Multiple correlation, where R^2 is the fraction of variation y explained by regression. The t-test is used to examine the significance of parameters.

1.4 Overview of the Thesis

Suppose d is the intrinsic dimension of the data set Z where $d \leq D$. The work described in this thesis develops algorithms for intrinsic dimension estimation methods in a nonlinear context. The core aim is to provide approaches for the estimation of intrinsic dimension. The proposed algorithms are based on the concept of charting manifolds [6] and on the correlation-dimension concept, detailed in ref [9].

The first part of this thesis represents an overview of existing methods of dimension reduction and intrinsic dimension. This thesis continues with our approaches towards ID estimation via correlation integral and charting manifold. The later chapters carry out the application of the algorithms on experimental data sets and adopt several methods to make comparisons. Various data examples are provided to illustrate the developed methodology, initially handling situations with intrinsic dimensionality equal to 1, and later proceeding to higher-dimensional examples. Furthermore, simulation examples are presented to study the efficiency of the algorithms. The algorithms also are implemented on recorded data from an industrial glass melter process provided by Dr. Uwe Kruger.

The chapters structure is as follows. Chapter 2 introduces briefly the concepts for linear methods; Principal component analysis, Independent component analysis, Linear discriminant analysis method and Principal variables. We also present the concepts for Nonlinear methods of dimension reduction; Nonlinear PCA, Principal curves and manifolds, Multidimensional scaling and ISOMAP, Locally linear embedding, Self-organising maps and Visualisation induced SOM. The relationship between intrinsic dimension and some dimension reduction methods is illustrated. Additionally we discuss the relationship between the algorithms and their computational cost.

Chapter 3 presents the concepts for local methods of dimensionality estimation methods; Fukunaga-Olsen's algorithm, The near neighbor algorithm, TRN-based methods, Charting a manifold method and the Maximum likelihood estimation. The concept of global methods of dimensionality estimation methods are explained; Projection techniques, Multidimensional scaling methods, and Fractal-based methods. We also provide the implementation results of some of ID methods on the artificial data sets. In addition, we present an overview of the intrinsic dimension estimation methods by exploring the computation costs and other factors.

Building on these concepts, Chapter 4 introduces the algorithms developed and the new approaches in this thesis. We also provide a discussion and illustration of the approaches. This is followed by contrasting these algorithms in Chapter 5, which summarize the application studies. We discuss the computational results for data sets in multivariate space, and the effectiveness of our techniques. Finally, Chapter 6 contains a concluding summary and suggested areas for investigation in the future work.

Chapter 2

Dimension Reduction Methods

2.1 Introduction

The aim of this chapter is to review the methods of dimension reduction. In many applications which deal with high-dimensional data sets the researchers found that not all variables are needed to represent the data. It is worth reducing the dimensionality into a lower dimension in order to analyze the data set more efficiently and accurately. This is done by using dimension reduction methods. Those techniques are often applied as a data pre-processing steps or as part of data analysis to simplify the data model. Dimension reduction techniques transform the data set Z from high-dimensional variable space (D) (embedding space) onto a new data set with low-dimensional space (d) (manifold space) such that $d \leq D$.

Dimension reduction methods can be classified as *linear* and *nonlinear* methods. *Linear methods* try to search a globally flat subspace such as principal component analysis and projection pursuit. The aim of most of these methods is to reduce dimensionality by a linear transformation of all original variables such as principal component analysis (PCA). The linear methods are most widely used due to their simplicity and are easier to compute and describe the mapping (representation). *Nonlinear methods* try to search a locally flat subspace, such as multidimensional scaling methods and ISOMAP. Usually nonlinear algorithms assume that the relationship between neighboring points holds more information than the information from the relation between distant points [58].

Section 2.2 discusses briefly existing dimension reduction methods using linear approaches. Section 2.3 presents an explanation of nonlinear dimension reduction methods. The relationship between the dimension reduction methods and the intrinsic dimension is discussed in Section 2.4. The comparison between *linear* and *nonlinear* methods is explored in Section 2.5.

2.2 Linear Methods

In this section the linear dimension reduction methods are briefly reviewed. Firstly principal component analysis method is explained in Subsection 2.2.1. Independent component analysis technique is illustrated in Subsection 2.2.2. Linear discriminant analysis method and principal variables method are presented in Subsection 2.2.3 and Subsection 2.2.4, respectively.

2.2.1 Principal Component Analysis

Principal Component Analysis (PCA) is an unsupervised feature extraction and the most popular linear technique. It is also known as a proper orthogonal decomposition or Karhunen Loeve transform in the machine learning literature. The PCA technique reduces the number of variables and uses those few variables to explain the significant information of the data set. The technique was first introduced by Pearson in 1901, by finding lines and planes that present a good fit for given points in multivariate data. Joliffe (2002) [48] developed an interesting illustration of PCA properties and applications. The PCA is obtained via linear approximation and decomposing variance techniques as following.

Linear approximation technique

Assume that X is a random vector, see Section 1.2, with mean m and variance Σ. Then X is approximated through a single straight line by minimizing the expected squared distances between X and their projection X' onto the line, i.e. minimize $E\left(\overline{XX'}^2\right)$ [21]. By Pythagoras,

$$E\left(\overline{XX'}^2\right) = E\left(\overline{mX}^2\right) - E\left(\overline{mX'}^2\right),$$

where $\overline{XX'}$ denotes the length of the line segment connecting X and X', i.e. $\|X - X'\|$. Here minimizing $E\left(\overline{XX'}^2\right)$ means maximizing $E\left(\overline{mX'}^2\right)$ which yielding to maximize var $\left(\gamma^T X\right)$ [21], then

$$\text{var}\left(\gamma^T X\right) = \lambda,$$

where $\gamma$is one of the orthogonal eigenvectors $\gamma_1, \cdots, \gamma_D$ of Σ, and λ is one of the D eigenvalues of $\Sigma \in \mathbb{R}^{D\times D}$ such that $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_D > 0$. Therefore the eigenvector γ_1 is chosen corresponding to the largest eigenvalue λ_1.

Now, $\gamma_1^T X$ is the new random variable which is a linear combination of X with maximal variance, it is also known as the first principal component of X. The corresponding first principal component line is defined as the line

$$y_1(p) = m + p\gamma_1, \quad (p \in \mathbb{R}),$$

where p is the Euclidean distance between m and x' for a particular point $x \in \mathbb{R}^D$ [21]. Similarly, the $j-$th principal component is given by $\gamma_j^T X$, and

$$y_j(p) = m + p\gamma_j$$

is the corresponding $j-$th principal component line. Note that the first principal component is self consistent, which means that any point on the line is the conditional expectation of X over the points of the space which project to this point [52]. The second principal component has the highest variance among all the linear combinations orthogonal to the first principal component, and so on. Figure 2.1 shows an example of Horse mussels data cloud with two variables and its principal components.

Decomposition of variance

An important characteristic of PCA is the decomposition of the variance of X. For $j-$th eigenvector γ_j of Σ, one has

$$\Sigma\gamma_j = \lambda_j\gamma_j \quad j = 1, \cdots, D,$$

which can be written as [21]

$$\Sigma\left(\gamma_1,\cdots,\gamma_D\right)=\left(\gamma_1,\cdots,\gamma_D\right)\begin{pmatrix}\lambda_1 & & \\ & \ddots & \\ & & \lambda_D\end{pmatrix}$$

Then,

$$\Sigma\Gamma \;=\; \Gamma\Lambda \tag{2.1}$$

$$\Sigma=\Gamma\Lambda\Gamma^{-1}=\Gamma\Lambda\Gamma^{T}. \tag{2.2}$$

This decomposition is called the eigen decomposition of Σ, we have

$$\lambda_j=\mathrm{var}\left(\gamma_j^{T}X\right),\quad \text{for }\; j=1,\dots,D,$$

which means that λ_j provide some decomposition of variance, and their sum [21] is:

$$\lambda_1+\cdots+\lambda_D=\mathrm{Tr}\left(\Lambda\right)\quad \text{from Eq.(2.1)}$$

Thus,

$$\lambda_1+\cdots+\lambda_D=\mathrm{Tr}\left(\Gamma^{T}\Sigma\Gamma\right)=\mathrm{Tr}\left(\Gamma^{T}\Gamma\Sigma\right)=\mathrm{Tr}\left(\Sigma\right)\equiv \mathrm{TV}(X).$$

The trace of the variance matrix is called the total variance. Therefore,

$$\frac{\lambda_j}{\lambda_1+\cdots+\lambda_D}=\frac{\mathrm{var}\left(\gamma_j^{T}X\right)}{\mathrm{TV}(X)},$$

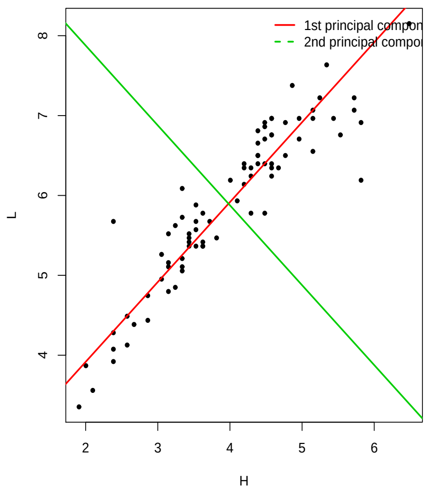


Figure 2.1: Principal component analysis from scaled Horse mussels data with two variables.

is the proportion of total variance explained by the $j-$th principal component [21]. Some software packages, such as R package [74], illustrate this decomposition by plotting λ_j versus j using the scree plot tool, more details in Subsection 3.4.1.

Assume that PCA has been carried out on a data set Z (see Section 1.2) yielding $m,\ \Sigma,\ \gamma_1, \cdots, \gamma_D,\ \lambda_1, \cdots,\ \lambda_D$. Now to compress the data Z to a smaller dimension $d \leq D$ means to project all data points (N) onto the d-dimensional subspace spanned by the d largest principal components:

$$\Phi : \mathbb{R}^D \to \mathbb{R}^d,\ x_i \mapsto (\gamma_1, \cdots, \gamma_d)^T (x_i - m), \quad i = 1, \cdots, N. \tag{2.3}$$

The $\Phi(x_i) \equiv p_i$ are called scores. It is obvious that the original data will not be reconstructed exactly unless $d = D$. PCA applications are found in many fields, such as pattern recognition [14], image processing [88], regression application [21] and data mining. Practically the core point of the PCA method is that the user needs to decide the number of components that reduce the variance. The methods to select the significant variables will be briefly discussed in the next chapter.

Despite its wide use, the PCA technique implies an assumption of linearity and can not capture nonlinear relationships of higher dimension than two [52] [95]. Those problems are solved efficiently with nonlinear PCA methods, such as local PCA and nonlinear PCA methods.

2.2.2 Independent Component Analysis

Jutten and Herault (1991) [44] [45] proposed Independent Component Analysis (ICA) as an approach for analyzing multivariate data. Independent component analysis is an unsupervised linear method. It reduces the dimension of a given data set by computing linear projections. The ICA algorithm has a facility which enables it to find the underlying components and sources that are mixed in the original data, where in many cases the classical methods failed to compute them [45]. The algorithm assumes that the components are independent and non-Gaussian. Hyvärinen et al. [44] provided a comprehensive explanation of ICA and its applications.

As for the PCA method, assume that the data Z is modelled as a linear combination of hidden variables s

$$x_i = \sum_{j=1}^{d} w_{ij}\, s_j, \quad \text{for } i = 1, \cdots, D, \tag{2.4}$$

where x_i are observed variables and both w_{ij} and s_j need to be estimated. Additionally, s_j are independent components while the coefficients w_{ij} are called the mixing coefficients. This estimation is also known as blind source separation [45].

Then, the model becomes

$$X = WS,$$

where X and S are random vectors, and W is an orthogonal matrix of parameters. The algorithm assumes the following [45]:

- The components s_i have non-Gaussian distributions.
- The components s_i are mutually statistically independent.
- The matrix W is $D \times D$ matrix.

ICA algorithm estimates the mixing matrix W based on a pre-whitening process, which means that the data is linearly transformed by a matrix A, such that $Y = AZ$ where the matrix Y has zero mean and identity covariance matrix [45]. Then the ICA model is

$$Y = AZ = AWS = \tilde{W}S. \tag{2.5}$$

Hence, the matrix $\tilde{W}$ is an orthogonal matrix, which reduces the number of free parameters in the model. The importance of whitening is illustrated by Hyvärinen [45]. For Gaussian variables whitening exhausts all the dependence information in the data. For non-Gaussian variables whitening does not imply independence.

Now the matrix $\tilde{W}$ is estimated by maximizing the ICA objective functions rather than the covariance matrix of Z. Note that the objective functions could be considered as high-order statistics, such as kurtosis and nonlinear correlations, which are used to determine the non-Gaussianity of components. Then an optimization method, such as the natural gradient method, is applied to optimize the objective function [44] [45].

2.2.3 Linear discriminant analysis

Linear discriminant analysis (LDA) is a supervised feature extraction method which is used to find the best separation between given groups. The LDA technique reduces dimensionality while preserving most of the information of the groups. It assumes that the data set is classified into two or more groups of objects. Similar to PCA and ICA, LDA attempts to to find a linear transformation with the best data representation. Furthermore, the technique considers the differences within-classes and differences between-classes [16] [67]. Compared to PCA, LDA keeps the original location of data points after the transformation [16] [67]. The LDA technique transforms the data set with verification of the separation in the data.

LDA was developed by Fisher in 1936. Fisher's LDA technique attempts to find a transformation that maximizes the differences between-classes S_B and minimizes the differences within class S_W. The maximization is called the Fisher criterion [30] [16] [67]. Now, suppose there are c classes, let m be the overall mean of the data, m_i be the mean vector of all samples in class i, and n_i be the number of samples in class i, where $i = 1, 2, \cdots, c$. The total number of samples is $N = \sum_{i=1}^{c} n_i$. By defining

$$S_B = \sum_{i=1}^{c} (m_i - m)(m_i - m)^T, \tag{2.6}$$

$$S_W = \sum_{i=1}^{c} \sum_{j=1}^{n_i} \left(x_j - m_i\right) \left(x_j - m_i\right)^T, \tag{2.7}$$

$$m = \frac{1}{c} \sum_{i=1}^{c} m_i, \tag{2.8}$$

LDA computes the ratio of the differences between-class and differences within-class, then one has

$$w_{LDA} = \frac{w^T S_B w}{w^T S_W w}, \tag{2.9}$$

where w_{LDA} is determined by the eigenvectors corresponding to the largest eigenvalues of $S_W^{-1} S_B$. Thus, there will be at most $c - 1$ non-zero eigenvalues [67]. In recent papers nonlinear generalizations of LDA are proposed such as Kernel Discriminant Analysis and Local Fisher Discriminant Analysis.

2.2.4 Principal Variables

Principal variables (PV) are a subset of the original data according to special criteria. This subset performs the best representation and preserves the information from the original variables. Consider a data matrix $Z \in \mathbb{R}^D$ consisting of N observations with sample covariance Σ and correlation matrix R. In the same manner of [15] [68], assume that X is partitioned into subsets (X_1, X_2) where X_1 consists of m vectors of retained variables and X_2 is a $(D - m)$ vector of discarded variables. Then the covariance matrix Σ is

$$\Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}, \tag{2.10}$$

where Σ_{11} is the $m \times m$ covariance matrix of X_1, and there are $2^D - 1$ choices of set selection for all $m = \ 1, \ \cdots \ , \ D$. The partial covariance matrix for X_1, given X_2, is

$$\Sigma_{22.1} = \Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}. \tag{2.11}$$

The partial correlation matrix $R_{22.1}$ is obtained by scaling $\Sigma_{22.1}$ which gives unit diagonal elements.

MacCabe(1984) proposed a number of optimal criteria for choosing principal variables (PV) selection [68]. He suggested 12 criteria which lead to the following solutions:

$$\text{M1. } \max |\Sigma_{11}| \equiv \min |\Sigma_{22\cdot 1}| \equiv \min \prod_i \lambda_i.$$

$$\text{M2. } \min tr(\Sigma_{22\cdot 1}) \equiv \min \sum_i \lambda_i.$$

$$\text{M3. } \min \|\Sigma_{22\cdot 1}\|^2 \equiv \min \sum_i \lambda_i^2.$$

$$\text{M4. } \max \sum_{i=1}^{k} \rho_i^2, \text{ with } k = \ \min(m, D - m),$$

where $|A|$ and $\text{tr}(A)$ are the determinant and the trace of the matrix A; $\|A\|^2$ is the squared norm $\left(\sum\sum a_{ij}^2\right)$; λ_i are the eigenvalues of $\Sigma_{22.1}$; and the ρ_i are the canonical correlations between the selected and unselected variables [15]. Stepwise selection is used to obtain the near-optimal subsets for M2 while for M1, M3, M4 the optimal subsets need to be evaluated for all possible subsets and become less easy to compute with large variables [15] [68].

In 2007, Cumming and Wooff [15] proposed an alternative criterion based on the spectral decomposition of the $(D \times D)$ correlation matrix. Assume that $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_D > 0$ are the ordered eigenvalues of R and $a_1, \cdots, a_D$ the associated eigenvectors. Then the correlation matrix R can be written as

$$R = \sum_{i=1}^{D} \lambda_i a_i a_i^T = A\Lambda A^T, \tag{2.12}$$

where A is a $(D \times D)$ orthonormal matrix with columns which are the a_i and Λ is $(D \times D)$ diagonal matrix with entries λ_i [15]. Now, similar to criterion M3, the $\|R\|^2$ can be written as:

$$\|R\|^2 = \sum_{i=1}^{D} \lambda_i^2 = \sum_{j=1}^{D}\sum_{i=1}^{D} (\lambda_i a_{ji})^2 = \sum_{j=1}^{D} \left(\sum_{i=1}^{D} r_{ij}^2 \right) = \sum_{j=1}^{D} h_j, \tag{2.13}$$

where

$$h_j = \sum_{i=1}^{D} r_{ij}^2 = \sum_{i=1}^{D} (\lambda_i a_{ji})^2 .$$

Therefore, the first principal component is a linear combination of the original variables. This component has a maximum contribution to $\|R\|^2$ while the remaining principal components giving less contribution. The values of h_j will be large when variable x_j has, on average, high loadings on important principal components [15]. Now, applying a stepwise algorithm for variable selection using the above criteria, the technique searches for the m variables with highest values of h_j, such that $\sum_{j=1}^{m} h_{(j)}$ makes some predetermined proportionate threshold [15]. The technique works as follows [15]: calculate the values of h for each variables x_j for $j = 1, \cdots, D$. Then select the best variables with the largest values of h_j and compute a partial correlation matrix for the remaining variables, and repeat the process. This iteration process makes sure that the chosen variable captures aspects of the variation which are not represented by the previously selected variables [15].

Cumming and Wooff [15] showed that the extension method is suitable for determining PV for repeated measures data, and it is also uncomplicated.

2.3 Nonlinear Methods

In this section the nonlinear dimension reduction methods are briefly covered. The nonlinear methods we illustrate in the subsections are used mainly for dimensionality reduction and less for intrinsic dimension estimation. Besides the principal curves and manifolds are not always suitable for all data structures. Firstly nonlinear PCA is explained in Subsection 2.3.1. Principal curves and manifolds are illustrated in Subsection 2.3.2. Multidimensional scaling and ISOMAP methods are presented in Subsection 2.3.3 and Subsection 2.3.4, respectively. In Subsection 2.3.5 an illustration of locally linear embedding method is presented. Self-organising maps and visualisation induced SOM are discussed in Subsection 2.3.6 and Subsection 2.3.7, respectively.

2.3.1 Nonlinear Principal Component Analysis

In 1980 the development of nonlinear PCA methods came under consideration. Those methods addressed the linearity constraints of PCA. Nonlinear PCA techniques can be divided into the utilization of autoassociative neural networks, principal curves and manifolds, and Kernel approaches. Kruger et al. [58] presented a review of existing nonlinear PCA techniques and also examined the needs of nonlinear PCA methods in practice. Kruger et al. [55] introduced a non linearity test that studies the structure (linear or nonlinear) of a given data set by analyzing the variables interrelationship. In the following the Kruger et al. [58] test is explained.

Firstly, the data operating region is partitioned into several disjoint regions, where the first region is centered around the coordinate system origin, then the PCA technique is applied on the data points of each region. In substance, dividing the operating region into the disjoint regions is computed through a prior knowledge of the process or by directly analyzing the recorded data. Using a prior knowledge into the construction of the disjoint regions, requires the incorporation of knowledge about distinct operating regions of the process [58]. In contrast, a direct analysis by plotting scatter plots of the first few retained principal components could expose patterns that are indicative of distinct operating conditions [58]. Practically, if the direct analysis does not yield any distinctive features, then the original operating region could be divided into two disjoint regions initially, and applying the nonlinearity test to these two disjoint regions. Besides the number of regions is increased incrementally and followed by a subsequent application of the test. Note that the increasing of the number of disjoint regions leads to reduce the number of observations in each region [58].

Finally the data structure is determined as follows. The accuracy bounds that are based on the residual variance are computed for one of the PCA models, and the residual variance of the remaining PCA models are benchmarked against these bounds [58]. The test is completed if each of the PCA models has been used to determine accuracy bounds which are then benchmarked against the residual variance of the respective remaining PCA models.

Therefore the data has a linear structure when each residual variance is within an accuracy bound. In contrast the data structure is nonlinear if at least one of the residual variances is outside the accuracy bound. Additionally, when the accuracy of the PCA model is smaller than the variation of the residual variances, one can conclude that the data structure is nonlinear [58]. Obviously the number of PCA models is equal to the number of disjoint regions. Kruger et al. [58] illustrated that this test is obtained under special assumptions and the reason for using the residual variance is because it is independent of the disjoint regions.

Nonlinear principal component algorithms have been proposed as an extension of PCA. The algorithms have been developed by Schölkopf et al. (1998) [27]. The following section reviews briefly nonlinear PCA techniques.

A. Autoassociative Neural Network Approach

In the early 1990s, Kramer [58] proposed a generalization of nonlinear PCA using an Autoassociative Neural Network (ANN). The ANN projects the recorded data onto a curve or surface [95]. The network consists of five layers: input layer, mapping layer, bottleneck layer, demapping layer and output layer, as displayed in Figure 2.2. The algorithm try to reconstruct the D network input variables using a reduced set of bottleneck nodes, i.e. the reduced variables $< D$.

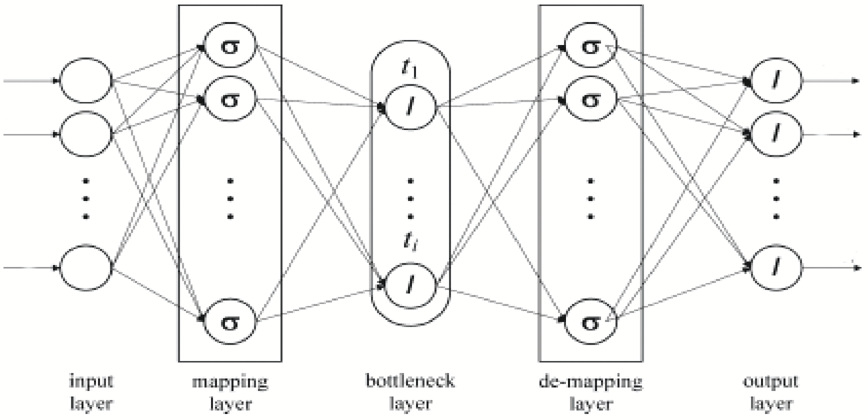


Figure 2.2: Autoassociative neural network layers (taken from [58]).

The input layer is the first ANN layer from the left, where weighted values of the original variable set Z are passed onto the second layer (mapping layer) [58]:

$$\xi_i = \sum_{j=1}^{D} w_{ij}x_j + b_i,$$

where w_{ij} are the weights for the first layer and b_i is a bias term. The algorithm produces nonlinear score variables in the middle layer, referred to as the bottleneck layer. The input of the fourth layer -demapping layer - is a linear combination of these nonlinear score variables. Finally, the nonlinear transformation provides the reconstruction of the original variables by the output layer [58].

The technique performs identity mapping, which means that the number of outputs of the network is equal to the number of inputs [58]. Then the outputs of the bottleneck layer, which is in the middle of the network, provide the nonlinear principal components. The number of necessary components is estimated by minimizing the squared distances of the data points using the first nonlinear principal component. The conjugate-gradient algorithm is an optimization algorithm of ANN and is generally used [8]. ANN is successfully used for analyzing climate data [5], and atmospheric and oceanic sciences data [79].

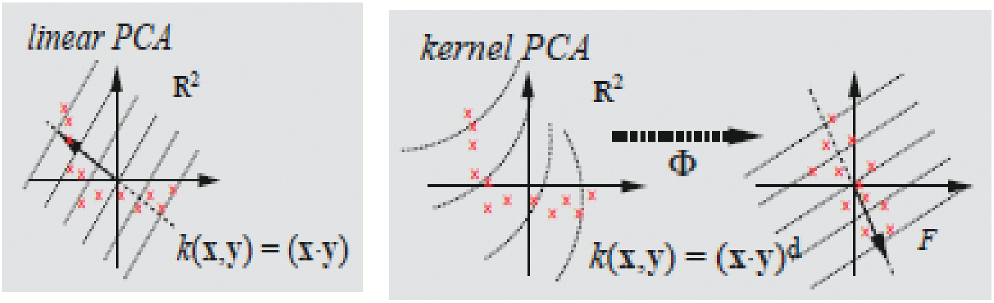
The technique is less effective for large data sets [8]. Other shortcomings are discussed by Kruger [58]. Scholz et al. [79] proposed a comprehensive illustration of autoassociative neural networks and studied the variants of networks with applications in the field of biology.

B. Kernel PCA

Kernel principal component analysis is a more recent nonlinear generalization of PCA. It is based on the use of the kernel function. The technique is proposed by Schölkopf et al. [58] [80]. In Kernel PCA the data $Z \in \mathbb{R}^D$ is mapped into a high-dimensional space, which is called the feature space, by a mapping function $\Phi(Z)$. Then, the algorithm performs a linear separation in that space and makes a nonlinear projection of the data set in a new space.
Thus

$Z \mapsto \Phi(Z)$, where $\Phi : \mathbb{R}^D \mapsto \mathbb{R}^M$,

and $M > D$ which means that $\Phi(Z)$ has a dimension considerably larger than D. Then the principal component analysis is performed on $\Phi(Z)$. Therefore, the data in the feature space is projected onto a low-dimensional subspace spanned by the eigenvectors which capture most of the variance. Figure 2.3 delineates the difference between linear PCA (Figure 2.3a) and Kernel PCA (Figure 2.3b). In (Figure 2.3b) the data points have a nonlinear pattern in the original space (left), while in the (right) the data points form a linear pattern in the high dimensional feature space.


linear PCA
$\mathbf{R}^2$
$k(\mathbf{x},\mathbf{y}) = (\mathbf{x}\cdot\mathbf{y})$
kernel PCA
$\mathbf{R}^2$
Φ
$k(\mathbf{x},\mathbf{y}) = (\mathbf{x}\cdot\mathbf{y})^d$
F

(a) (b)

Figure 2.3: (a) The data points are projected using the linear PCA method, (b) Kernel PCA, the data points in the original space are mapped into feature space by the mapping $\Phi(Z)$ (taken from [80]).

Now, following the explanation of the algorithm in [58], suppose that; $\Phi(Z) = [\Phi(x_1)\Phi(x_2)\ldots\Phi(x_N)]^T$ is the original centered feature matrices. Kernel PCA tries to compute

$$\Sigma_\Phi \gamma_i = \frac{1}{N-1}\bar{\Phi}(Z)^T\bar{\Phi}(Z)\gamma_i = \lambda_i\gamma_i, \quad \text{where } i = 1, \ldots, D, \tag{2.14}$$

where $\bar{\Phi}(Z) = \Phi(Z) - \frac{1}{N}E_N\Phi(Z)$, with E_N being a matrix of ones, is the original centered feature matrix .

In contrast, it is difficult to extract the eigenvectors directly from the the covariance matrix of $\Phi(Z)$ because $\Phi(Z)$ is an unknown formulation [58]. Therefore the formulation of the kernel function is used to overcome this deficiency.

Hence, suppose $G = \bar{\Phi}(Z)\,\bar{\Phi}(Z)^T$ and is further defined as the Gram matrix [58]:

$$\bar{\Phi}(Z)\,\bar{\Phi}(Z)^T v_i = \zeta_i\, v_i, \tag{2.15}$$

where ζ_i and v_i are the eigenvalue and its eigenvector, respectively. Now, by multiplying (2.15) by $\bar{\Phi}(Z)^T$, then

$$\bar{\Phi}(Z)^T\,\bar{\Phi}(Z)\,\bar{\Phi}(Z)^T v_i = \zeta_i\,\bar{\Phi}(Z)^T v_i, \qquad \text{for } i = 1, \ldots, D. \tag{2.16}$$

By comparing (2.14) and (2.16), it now follows that $\zeta_i/(D{-}1)$ and $\bar{\Phi}(Z)^T v_i / \left\|\bar{\Phi}(Z)^T v_i\right\|^2$ are also corresponding eigenvalues and eigenvectors of Σ_Φ, that is:

$$\lambda_i = \zeta_i/(D-1),$$

$$\gamma_i = \bar{\Phi}(Z)^T v_i/\sqrt{\zeta_i}.$$

Now, the kernel function is defined as $k(x_i, x_j) = \Phi(x_i)^T\Phi(x_j)$, and the Gram matrix G can be constructed from a kernel matrix $K(Z) \in \mathbb{R}^{N\times N}$ as [58],

$$G = K(Z) - \frac{1}{N}K(Z)\,E_N - \frac{1}{N}\,E_N\,K(Z) + \frac{1}{N^2}\,E_N\,K(Z)\,E_N.$$

Note that the calculation of G is depending only on $k(x_i, x_j)$. The most commonly used kernel functions include polynomial, RBF and Sigmoid kernels [58].

In addition, the data points represented in the kernel matrix are assumed to be centered in the feature space. The kernel matrix is a symmetric matrix with $N \times N$ and its elements are defined by the inner product of all pairs of points $\Phi(x_i)$ and

$\Phi(x_j)$, where $i, j = 1, \ldots, N$, in the feature space [58]. Then the reduced dimension is obtained by computing the eigenvectors of the kernel matrix. The score variables are derived such that the first one possesses a maximum variance, and the second largest variance and so on [58].

The computational demand for this technique increases insignificantly for large values of N. The drawback of the method is that it is dependent on the kernel choice. Besides, it is necessary to neglect the eigenvalues whose magnitude is lower than a threshold value that can only be fixed in a heuristic way [8]. Several papers discuss the comparisons between several techniques of nonlinear PCA [58] [95].

2.3.2 Principal Curve and manifolds

Principal Curve (PC) is a nonlinear generalization of PCA created by forming an embedded manifold, and by using standard geometric projections on the manifold. This technique is known as a nonparametric smoothing method. The principal curve is a smooth one-dimensional curve passing through the middle of a data cloud. Additionally, it can be considered as a one-dimensional manifold embedded in high dimensional data space [58]. Hastie and Stuetzle [38] [39] proposed this curve to approximate a one-dimensional nonlinear topological relationship of data points, which is usually two variables. Their definition is based on the notion of self consistency. Every point lying on the principal curve is the average (conditional mean) of all the data points that are projected onto it [55].

Consider the data matrix Z in $D-$dimensional space, where f is a smooth curve in $\mathbb{R}^D$ parametrized by $\lambda \in \mathbb{R}$. Let $\lambda_f(x)$ denote the value for which $f(\lambda)$ is closest to x [38] [39]. The projection index $\lambda_f(x)$ is defined by

$$\lambda_f(x) = \sup_{\lambda} \left\{ \lambda : \|x - f(\lambda)\| = \inf_{\mu} \|x - f(\mu)\| \right\}, \tag{2.17}$$

where $\|.\|$ denotes the Euclidean norm in $\mathbb{R}^D$.

Following Hastie and Stuetzle's definition, a principal curve has the following properties [52]:

- f does not intersect.

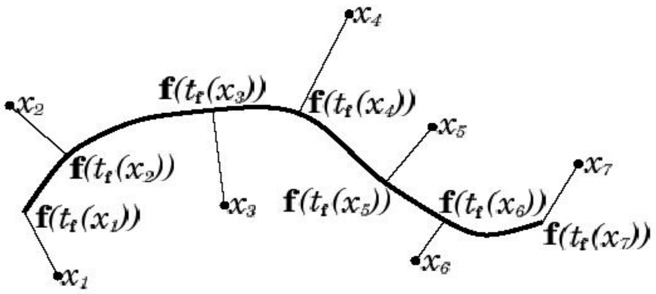


Figure 2.4: Projecting points to a curve (taken from [52]).

- f has a finite length inside any bounded subset of $\mathbb{R}^D$.
- f is self-consistent, i.e. $E\left(Z \mid \lambda_f(Z) = \lambda\right) = f(\lambda)$.

Various algorithms developing the *PC* technique have been proposed, such as Hastie and Stuetzle's algorithm for constructing principal curves, abbreviated as $\mathbf{HSPC}_s$ for a given data distribution [58]. Cubic smoothing splines and kernel smoothing can be used as a smoothing technique for the estimation of $\mathbf{HSPC}_s$ [58].

The principal curve algorithms can be divided into two families ('top-down') or ('bottom-up'), see Einbeck et al. (2005) [27] . The 'top-down' algorithms start with the first principal component of the data set as an initial line, then bend this line until the resulting curve passes satisfactorily through the middle of the data, and minimizes various global error criterion. However, in some cases the selection of an initial line leads to some technical problems and inflexibility, such as bias. There are various ways of tackling and solving this problem. For example, instead of starting with a global initial line, another option is to look exclusively the data in a local neighborhood for points in every step [26] [23]. This way the principal curve is constructed in a 'bottom-up' manner. Local principal curve (LPC) is one of the 'bottom-up' algorithms. It proceeds through the data and does not minimize a global error criterion [27]. In the next section we demonstrate the LPC technique.

Local Principal Curve

When we consider a data set Z with $x_i = (x_{i1}, \cdots, x_{iD})^T$, $i = 1, \cdots, N$. The idea of the algorithm is to seek a smooth curve passing through the middle of the data cloud, where the curve is obtained by computing local centers of mass of the data. This concept follows the proposed work of Einbeck et al. [27]. Figure 2.5 displays the Hastie and Stuetzle principal curve and local principal curve on Spiral data. The Local Principal Curve (LPC) algorithm works using the following steps [27]:

- Step 1: Choose a starting point $x = x_0$ which is in or close to the data cloud. This is done by choosing the point with the highest density or select it randomly.

- Step 2: Compute a local mean μ^x around x, where μ^x is given by $\mu^x = \sum_{i=1}^{N} w_i^x x_i$, and $w_i^x = \frac{K_H(x_i - x)x_i}{\sum_{j=1}^{N} K_H(x_j - x)}$ denotes an appropriate (bell–shaped) weight function centered at $x \in \mathbb{R}^D$, where H is a bandwidth matrix and $K_H(.)$ a $D-$dimensional kernel function.

- Step 3: A local principal component analysis is fitted at x by computing the first local eigenvector γ^x of $\Sigma^x = \left(\sigma_{jk}^x\right)$, $j \geq 1,\ k \leq D$, and

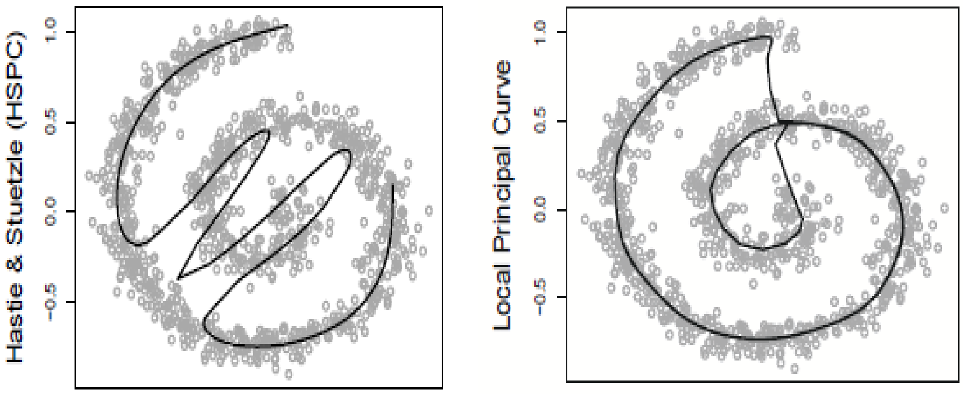
$$\sigma_{jk}^x = \sum_{i=1}^{N} w_i^x \left(x_{ij} - \mu_j^x\right)\left(x_{ik} - \mu_k^x\right),$$

  where μ_j^x is the $j-$th component of μ^x. Using z as step size, then step from μ^x to $x := \mu^x + z\gamma^x$.

- Step 4: Calculate the local mean μ^x.

Steps 3 and 4 are repeated until the algorithm produces approximately constant values of μ^x. Then the results of series μ^x are connected through a cubic spline and parametrized by its arc length. The series provide the local principal curve. Therefore every data point is projected to its nearest point on the curve, and the data is compressed corresponding to its projection index [27].

Principal curve algorithms provide a good representation for a given set of data, with the minimum dimension closest to one. *PC* is used in different applications, for



(a)

(b)

Figure 2.5: HSPC and LPC are obtained for the Spiral data (taken from [23]).

instance speech recognition, freeway traffic streams and the identification of profiles of ice floes in satellite images. The principal component can be considered as a special case of principal curves when the recorded data has an ellipsoidal distribution [95].

Local Principal manifold

Local Principal manifold (LPM) is a generalization of LPC algorithms proposed by Einbeck et al. [24]. The algorithm produces a representation of low-dimensional latent structures which could be used for data sets with 2 $\leq$ minimum dimension $\leq D$ (Einbeck et al. [24]). Applications of LPM algorithm are used for density estimation and classification on the manifold, and can also be used for studying the regression problem. An extension of the LPM algorithm is a local principal surface (LPS) which estimates a manifold of dimension $d = 2$. Further details on this technique are found in Einbeck et al. (2010) [26].

The LPM steps 1 and 2 are similar to the LPC steps outlined above, as illustrated in [24], and then

- Step 3: By extrapolating triangular surface, compute the direction of the vector that connects to the previous and current μ^x.

- Step 4: Adjust the principal curve towards the middle of the local data distribution via a constrained local mean.

This algorithm is used for the data set where the minimum dimension equals 2.

2.3.3 Multidimensional Scaling

Multidimensional Scaling (MDS) is a nonlinear projection technique that projects data points onto a two-dimensional manifold. MDS tends to provide a representation of distance and similarity patterns among data sets. The technique attempts to project the data set in such a way that preserves the pairwise distances between data points [8]. A general fitness function or stress function is defined as [95]

$$S = \frac{\sum_{i,j} (d(x_i, x_j) - D(x_i, x_j))^2}{\sum_{i,j} (D(x_i, x_j))^2}, \tag{2.18}$$

where $d(x_i, x_j)$ is the dissimilarity of data points i and j in the original data space, $D(x_i, x_j)$ is the distance (usually Euclidean) between mapped points i and j in the projected space (new space).

MDS maps the data with the least stress possible using an optimization algorithm. Several methods of MDS with different cost functions and optimization algorithms exist. The common algorithm used for this family is a gradient method. When the stress value equals zero then a suitable mapping (projection) is obtained. The well known stress function is proposed by Kruskal and Shepard [8] [72] and is defined as

$$S_{Kruskal} = \left[\frac{\sum_{i<j} [\text{rank}(d(x_i, x_j)) - \text{rank}(D(x_i, x_j))]^2}{\sum_{i<j} \text{rank}(D(x_i, x_j))^2} \right]^{\frac{1}{2}}. \tag{2.19}$$

Bennett's algorithm and Sammon's mapping are MDS methods that are closely related to Kruskal and Shepard's algorithm [8] [95] [72]. Bennett's algorithm assumed that the data has a uniform distribution in the sphere of radius r. The inter-point distance (Euclidean distance) between two data points is computed [72] as

$$E = \frac{|x_1 - x_2|}{2r}.$$

Then the variance of E is a decreasing function of the dimension D and could be expressed [72] as

$$D \cdot \mathrm{var}(E) \approx \text{constant}$$

which means that the increasing of variance E will flatten the data set. The algorithm works [8] as follows. Firstly, the patterns are moved to increase the variance of E. Secondly, the position of the patterns is adjusted which makes the rank orders of E the same in local regions. The process is iterated several times until the variance of inter-point distances levels off. Then the covariance matrix of the data set is computed, and the number of eigenvalues is obtained. This method tends to overestimate the intrinsic dimension of a data set, and it also needs to fix a threshold value. Fukunaga-Olsens algorithm faces a similar issue when it determines the retained eigenvalue [8].

Sammon's mapping is similar to Kruskal and Shepard's algorithm where the stress is minimized by the gradient–descent algorithm. Sammon's stress is

$$S_{Sammon} = \left[\sum_{i<j} \frac{\left(d(x_i, x_j) - D(x_i, x_j)\right)^2}{d(x_i, x_j)} \right] \left[\sum_{i<j} d(x_i, x_j) \right]^{-1}. \tag{2.20}$$

In practice, with Kruskal's method and Sammon's method the stress is minimized by moving all points simultaneously in the output (mapping) space [8] [72]. Another stress function has been proposed by Chang et al. [12] which improved Kruskal's method and Sammon's method. Chang's method tries to minimize the stress by moving two points at a time, which preserves the local structure. The issues with this method are that it needs high computation resources, even for a moderate number of data points. Furthermore, the results of Chang's method are influenced by the order in which the data points are taken as a pair [8] [72].

There are several other issues with the MDS method as follows [8] [95]:

- MDS is computationally intensive.
- It is difficult to display and analyze the data in a high-dimensional space.
- For each new set of data points the technique needs to compute every data point again.

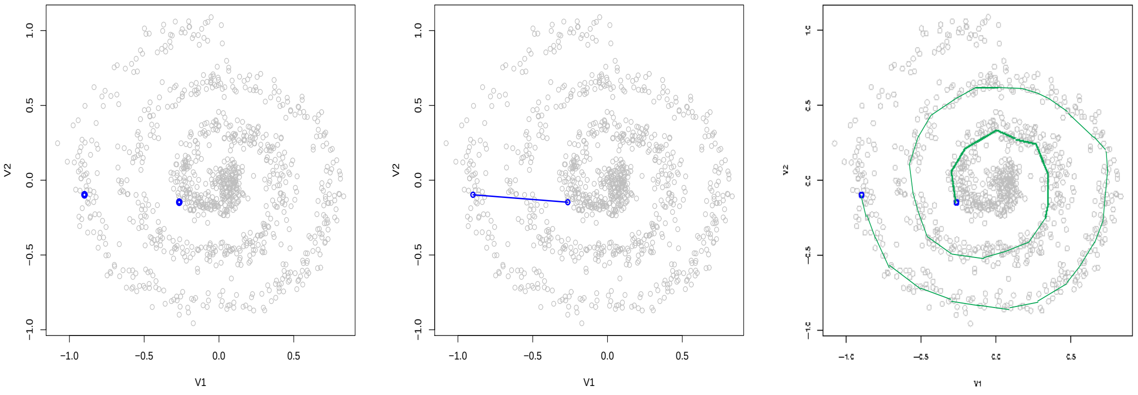
The technique is widely used in the applications of visualization and data mining in fields such as marketing and ecology.

2.3.4 ISOMAP

Isometric feature mapping method (ISOMAP) is a nonlinear method. It has been proposed as an extension of metric MDS. Fundamentally ISOMAP uses geodesic manifold distances between all data pairs instead of the Euclidean distance. Figure 2.6 displays the illustration between Euclidean and geodesic distance.

The technique was proposed by Tenenbuam et al.(2000) [84]. The ISOMAP algorithm tries to construct a low-dimensional embedding of a set of data points lying in high-dimensional space.

The technique used the input-space distances to estimate the geodesic distance between distant points [84]. The ISOMAP algorithm, as explained in [84], works as follows:



(a) (b) (c)

Figure 2.6: The difference between Euclidean and geodesic distances explained by two points in a spiral of two-dimensional space (based on Lee et al. (2004) [59]). (a) shows data points, (b) shows the Euclidean distance between the two points, (c) shows the geodesic distance between them is the same as along the manifold, which illustrates the intrinsic similarity of two points.

A graph G is constructed by connecting all neighbouring points and labellings all arcs with the Euclidean distance between the corresponding points, so the graph edges are between neighbours and distance weights. Next, the geodesic distance between two points is approximated by the sum of the arc lengths along the shortest path connecting both points. Several algorithms are proposed to compute the shortest paths, such as the algorithm of Tenenbaum, where the algorithm exploits the sparse structure of the neighbourhood graph [84]. The final step of the ISOMAP algorithm is to apply classical MDS to the approximated geodesic distance matrix, which means computing their largest eigenvectors. The eigenvectors provide the coordinates of the data points in the lower-dimensional space.

ISOMAP produces globally optimal mapping which is low-dimensional compared to PCA and MDS. Increasing the sample size provides a better approximation of the intrinsic geodesic distances [84].

2.3.5 Locally Linear Embedding

The Locally Linear Embedding method (LLE) is an unsupervised learning algorithm. Both the LLE and ISOMAP methods are known as a new generation of dimension reduction methods. The LLE algorithm has been proposed by Roweis and Saul (2000) [76]. It has several advantages over ISOMAP, including an ideal method to preserve the local geometry structure of the data. The LLE technique determines every data point and its k-neighbors, then uses the same weights to compute the low-dimensional embedding.

Consider data consisting of N real-valued vectors x_i, each of dimensionality D, and they lie on or near a smooth nonlinear d-dimensional manifold with $d << D$. The aim is to map the high dimensional coordinates to low dimensional global internal coordinates on the manifold. In the same manner as Roweis and Saul (2000), the algorithm works using the following steps [76]:

- Step 1: Assign the neighbors of each data point x_i. To do this, calculate the Euclidean distances between all data points, and for each data point select the k nearest neighbors.

- Step 2: Calculate the weight matrix W, where w_{ij} summarizes the contribution of the jth data point to the ith reconstruction. Measure the reconstruction errors using the following cost function:

$$\epsilon(w) = \sum_{i}^{N} \left| x_i - \sum_{j}^{N} w_{ij} x_j \right|^2 . \tag{2.21}$$

  The cost function Eq.(2.21) is governed by two restrictions: first, data points in x_i are reconstructed from its neighbors (i.e. $w_{ij} = 0$ when x_j not belongs to neighbor of x_i). Second, the sum of weights equal to one (i.e. $\sum_j w_{ij} = 1$). Then use a Lagrange multiplier to minimize the reconstruction error.

- Step 3: Map each x_i to a low-dimensional (embedded coordinates) y_i in global internal coordinates on the manifold. This mapping is achieved by minimizing the following cost function,

$$\Phi(w) = \sum_{i} \left| y_i - \sum_{j} w_{ij} y_j \right|^2 . \tag{2.22}$$

  In this final step, the algorithm reconstructs the local geometry represented by the weight matrix W in low-dimensional Euclidean space.

LLE has been applied to various applications, such as images of lips and facial expressions [76]. LLE works well with other methods in data analysis and statistical learning, and also the method achieves efficient computation.

2.3.6 Self-Organising Maps

Self-Organising Maps (SOM) is an unsupervised learning algorithm. The SOM tends to provide a representation of similarity patterns among a data set. The Kohonens Self-Organising Map proposed by Teuvo Kohonen is the most common model of a neural network. The technique attempts to project the data set in such a way that preserves the distances between data points as much as possible. It is also known as the topology preserving mapping of the original data space. Therefore the data points that are closest to each other in the original data space $\mathbb{R}^D$ are mapped to nearby neurons (nodes) in the new space [94] [95].

The SOM consists of a set of neurons that are arranged in a low-dimensional rectangular or hexagonal grid, to form a discrete topological mapping of an input space. In the same manner as Yin [95] described the algorithm, suppose the number of neurons equals m. w_{zi} is the weight vector of dimension D and associated with neuron i, where zi is the location vector of neuron i on the grid and $i = 1, 2, \cdots, m$. In the beginning of the learning, all the weights $\{w_{z1}, w_{z2}, \cdots, w_{zm}\}$ are initialized to small numbers randomly. Hence, following the illustration of the algorithm in [95], the SOM algorithm works as follows:

- Step 1: Determine the input $x(t)$, where $x(t)$ is an arbitrarily chosen element of data Z, and the winner for any time t,

$$v(t) = \arg\min_{a \in \Xi} \| x(t) - w_a(t)\| \text{, where } \Xi \text{ is the set of neuron indexes.} \tag{2.23}$$

- Step 2: The neighbors of the winner and their weights is updating as,

$$\Delta w_a(t) = \alpha(t)\ \eta(v, a, t)\ \left[x(t) - w_v(t)\right]. \tag{2.24}$$

- Step 3: The process is repeated until the map converges.

where $\eta(v, a, t)$ is the neighborhood function and could be a Gaussian function, i.e. $\eta(v, a, t) = \exp\left[-\frac{\|v-a\|^2}{2\sigma(t)^2}\right]$, and σ is the changing effective range of the neighborhood. The coefficients $\{\alpha(t), t \geq 0\}$ are scalar learning rate and monotonically decreasing, and satisfy [95]

1. $0 < \alpha(t) < 1$,
2. $\lim_{t\to\infty} \sum \alpha(t) \to \infty$,
3. $\lim_{t\to\infty} \sum \alpha^2(t) < \infty$.

Now, if the inner product similarity measure is used,

$$v(t) = \arg\min_{a \in \Xi} \left[w_a^T x(t)\right],$$

then the corresponding weight updating will become [95]:

$$w_a(t+1) = \begin{cases} \frac{w_a(t)+\alpha(t)x(t)}{\|w_a(t)+\alpha(t)x(t)\|} & a \in \eta_v \\ w_a(t) & a \notin \eta_v \end{cases}$$

This form is often used in text and document mining applications [95]. The SOM is used in many applications such as data visualization, clustering and classification. The drawback of the SOM is that the algorithm needs to mark the distance between neurons [95].

2.3.7 Visualisation induced SOM

Visualisation induced SOM (ViSOM) is the generalization (extension) of the SOM. It is proposed by Yin [94] [95] to overcome the drawbacks of the SOM. The method tries to preserve the inter-neurons distances on the map, by placing the nodes uniformly and smoothly in the nonlinear manifold. Therefore the distances will be the same between any two neighboring neurons, and the map will be a smooth manifold embedded into the data space [95]. Although the structures of the ViSOM and SOM are similar, the ViSOM method helps preserve a local inter-neuron distance on the map [95].

The ViSOM algorithm works, as illustrated in [95], by decomposing $x(t) - w_a(t)$ into two elements $[x(t) - w_v(t)] + [w_v(t) - w_a(t)]$, where the first element illustrates the updating force from the winner v to the input $x(t)$, and the second element is a lateral contraction force where neighboring neuron a is brought to the winner v. The lateral contraction force is constrained or regulated in order to help maintain a unified local inter-neuron distance $||w_v(t)w_a(t)||$ on the map [95]. One has the update rule

$$\Delta w_a(t+1) = \ w_a(t) + \alpha(t)\ \eta(v,a,t)\ \ \{[x(t) - w_v(t)] + \beta\,[w_v(t) - w_a(t)]\}\,. \tag{2.25}$$

such that

$$\beta = \frac{d_{va}}{D_{va}\rho} - 1,$$

where d_{va} is the distance of neuron weights in the input space, D_{va} is the distance of neuron indexes on the map, and ρ is a (required) resolution constant [95]. The contraction force is computed such that the distances between the nodes on the map are analogous to the distances of their weights in the data space [95]. The ViSOM algorithm tries to adjust inter-neuron distances on the map in proportion to that of the original space, so $D_{va}\rho \propto d_{va}$ [95].

Compared to Sammon mapping, the ViSOM preserves the original space as Sammon mapping and deals with training data and new input data points in a simple computational way [94]. Therefore the visualisation will be more direct, quantitatively measurable, and visually appealing. In addition, the map resolution may be developed by interpolating a trained map or incorporating local linear projections [95].

The SOM and ViSOM are similar in cases when the data is distributed uniformly, and also when the number of nodes becomes very large, in which case both the SOM and ViSOM will closely approximate the principal curve/surface [95].

2.4 The relationship between intrinsic dimension and dimension reduction

Dimension reduction describes the structure of complex data (explicitly or implicitly) through a small but sufficient number of variables. Most dimension reduction methods require the intrinsic dimension of the low-dimensional subspace to be fixed in advance. The intrinsic dimension (ID) is defined as the minimum number of variables which are necessary (suffice) to describe the data without much loss of information. For illustration consider the Spiral data with a two-dimension space, as in Figure 2.5. Consider also the principal curve which is a smooth one-dimensional curve passing through the middle of a data cloud, as shown in Subsection 2.3.2. In order to fit the principal curve to the Spiral data, the user has firstly to decide that the ID is equal to 1, as displayed in Figure 2.5.

Next, we demonstrate the relationship between intrinsic dimension and some of dimension reduction methods. For linear dimension reduction methods, such as principal component analysis (PCA), the data Z is compressed to a smaller dimension $d \leq D$. This means projecting all data points (N) onto the d-dimensional subspace spanned by the d largest principal components, as shown from 2.3 in Subsection 2.2.1. Then PCA reveals implicitly the intrinsic dimension estimate during the dimension reduction process. The number of significant variables represents the estimate of intrinsic dimension. For linear discriminant analysis (LDA) and independent component analysis (ICA) methods, as shown in Subsections 2.2.2 and 2.2.3, the user has to determine, in a similar way to PCA method, the eigenvectors corresponding to the largest eigenvalues. This step leads to estimate the intrinsic dimension.

For application of the principal curve method, the user needs to deduct firstly that the intrinsic dimension equals 1. Additionally the local principal manifold (LPM) which is an extension of principal curves, as shown in Subsection 2.3.2, produces a low-dimension representation and is used for the data where the minimum dimension equals 2. Then the user should decide that the ID equals 2 before fitting the local principal manifold. With the ANN method, the algorithm projects the recorded data onto a curve or surface, as shown in Subsection 2.3.1. In this method, the user needs to decide the dimension of the output space, fixing ID=1 or 2, as pre-processing step, before fitting the algorithm.

The multidimensional scaling (MDS) method projects data points onto a two-dimensional manifold, as shown in Subsection 2.3.3. This means that in the beginning the user sets the ID as equal to 2. On the other hand, its generalization method ‘ISOMAP’ produces globally optimal mapping, which is low-dimensional compared to PCA and MDS. ISOMAP constructs a low-dimensional embedding of a set of data points lying in a high-dimensional space, as shown in Subsection 2.3.4. The user should decide the dimensionality d of the manifold before applying the ISOMAP method. The most common setting is at ID=2.

The LLE method assumes the data points lie on or near the smooth nonlinear d-dimensional manifold with $d << D$, as in Subsection 2.3.5. The LLE method aims to map the data from high dimensional coordinates to low dimensional global internal coordinates on the manifold. In this case the user needs to know the dimension d of the manifold at the beginning.

To sum up, most dimension reduction methods require an explicit definition of the intrinsic dimension of the manifold. There have been few attempts dedicated to determining the estimate of the intrinsic dimension of data in this context.

2.5 Conclusion

In this section, we have given an overview of the methods of dimension reduction by exploring the relationship between the algorithms and their computational cost.

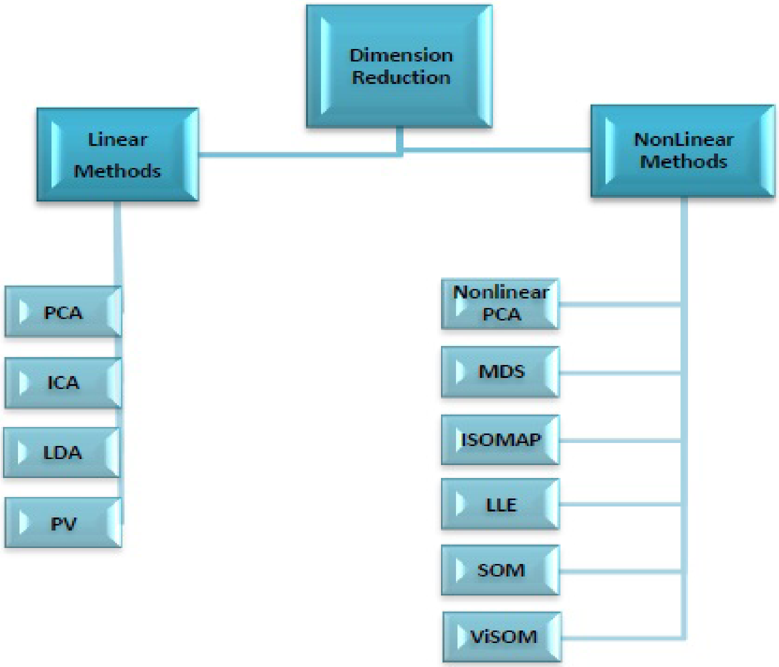


Figure 2.7: Dimension reduction methods

Figure 2.7 displays a taxonomy of techniques for dimension reduction which delineates that the core distinction between techniques is *linear* and *nonlinear* methods. Linear methods assume that the data set has a linear structure and the methods try to search for globally flat subspaces. Nonlinear methods for dimension reduction try to search for locally flat subspaces, and are not dependent on the assumption of linearity. The methods are used to embed the data in low-dimensional space.

Several other approaches to dimension reduction have been proposed. It is worth mentioning Laplacian eigenmaps (Belkin and Niyogi [3]) and Hessian eigenmaps (Donoho and Grimes [17]) which are motivated by spectral theory in the continuum. Laplacian eigenmaps are the predecessor of the next method – Hessian eigenmaps, which overcome the convexity limitation.

Next, the relationship between the algorithms and their computational cost is discussed. Firstly, the algorithms relationship is assessed. It become clear that several algorithms examined in Section 2.2 and 2.3 are related to each other. For instance linear PCA is a special case of the Kernel PCA with a linear kernel. ISOMAP is a special case of MDS which uses geodesic distances. Furthermore MDS is a special case which uses ISOMAP with k (number of nearest neighbors) equal to $N-1$.

Secondly, the computation cost is explored. Practically the computation cost and the method's memory capacity are determined by looking at the data properties, such as the original data set dimensionality D and the the number of data points N. Usually increasing N or even D leads to increase the computational cost proportionally. The computational cost is shaped by the number of parameters in the technique and the number of times iteration is needed. Most of the nonlinear methods have parameters which need to be optimized, for instance techniques that are based on neighbors such as ISOMAP and LLE. In addition to the technique's parameters, the nonlinear methods have higher computation costs than the linear methods, although this is outweighed by improvements in performance.

Chapter 3

Estimation Methods of Intrinsic Dimension

3.1 Introduction

This chapter introduces intrinsic dimension (ID) and examines the methods that are used to estimate it. The estimation of intrinsic dimension is an essential step in the dimension reduction process, because most dimension reduction methods require the intrinsic dimension of the low-dimensional subspace to be fixed in advance. When this is done the researcher can then deal with a space with a much lower dimension than the dimension of the original data set, such as a nonlinear manifold. Ideally the dimension should be reduced in a way which captures significant information embedded within the data set.

The word dimension has various definitions such as topological, intrinsic, fractal, and manifold dimension. These dimensions can be estimated for data sets. The d-dimensional manifold is a D-dimensional space $\mathbb{R}^d$ with dimension d [78]. The topological dimension of a topological space is either defined as the set of dimension D which can be divided into small sets as efficiently as possible, or as the dimension of the manifold that the data lies on [78].

Hausdorff dimension d_H - this is the first definition of a dimension [78] [8]. The D-dimensional Hausdorff measure $\Gamma_H(r)$ of a set is defined as:

$$\Gamma_H(r) = \lim_{r \to 0} \inf_{s_i} \sum_i (r_i)^D, \tag{3.1}$$

where the set is covered by small sets (cells) s_i with variable diameter r_i , such that all diameters satisfy $r_i < r$. The D-dimensional Hausdorff measure generalizes the usual notion of the total length, area and volume of simple sets [8]. Hausdorff [8] proved that

$$\Gamma_H(r) = \begin{cases} +\infty & \text{if} \quad D < \text{some critical value } d_H \\ 0 & \text{if} \quad D > d_H \end{cases},$$

where the critical value d_H is defined as the Hausdorff dimension of the set.

The definition of intrinsic dimension is delineated in the Section 3.2. This chapter is organized as follows. Section 3.3 briefly defines *local concepts*. An implementation of one of the local methods on artificial data sets is presented in Subsection 3.3.6. The main features of *global concepts* are briefly introduced in Section 3.4. An implementation of one of the global methods on artificial data sets is discussed in Subsection 3.5.2. An overview of intrinsic dimension estimation methods based on an exploration of computation costs and other factors is presented in Section 3.6.

3.2 Intrinsic dimensionality techniques

Assume the intrinsic dimension (ID) of a data set Z is given by a value d where $d \leq D$, which effectively captures the minimum number of variables necessary to describe the data without much loss of information [8] [32]. Camastra illustrated that the ID$= d$ is obtained when the data points lie entirely within a d-dimensional linear subspace of $\mathbb{R}^D$ [8]. This 'linear ID' is extracted by linear methods such as PCA, Factor analysis and Independent component analysis. Fukunaga's notion of ID [32] is as follows:

> "The geometric interpretation is that the entire data set lies on the topological curve of d or less dimensions."

This motivates the nonlinear techniques used in Fukunaga-Olsen's algorithm, Multidimensional scaling and fractal based methods. Following this concept, we have in this chapter a general notion of 'subspace' in mind which comprises linear as well as nonlinear manifolds.

Several papers such as Levina and Bickel [60] categorize the methods for the estimation of intrinsic dimension into two different groups, which are *projection techniques* and *geometric approaches* [60]. Following Camastra's survey [8], ID estimation methods can be classified into two groups. *Local methods* divide the data set into small subregions, or provide a series of local ID estimates at several target points, in order to arrive at a suitably averaged overall ID estimator. Examples to such methods include Levina–Bickel's maximum likelihood estimator [60], and Brands' concept of 'charting' [6], among others [8]. On the other hand, *global methods* try to estimate the dimension using the whole data set, imposing the implicit assumption that the intrinsic dimension is constant over the data set. Examples to such methods include projection methods, MDS and fractal-based method. The core aim of ID methods is to capture significant information that is embedded within the recorded set. Figure 3.10 illustrates the relationship between local, global, linear, and nonlinear ID methods.

Next assuming the relationship between the variables of a given data set are defined by a general model which describes, for each $x \in \mathbb{R}^d$ generated by a random vector X, a linear form

$$x = \mathbf{A}\mathbf{s} + \Delta x, \tag{3.2}$$

then following this linear model, one can define the nonlinear model as

$$x = \psi\left(\mathbf{s}\right) + \Delta x. \tag{3.3}$$

Here, x and $\Delta x \in \mathbb{R}^D$ while $\mathbf{s} \in \mathbb{R}^d$. The general assumptions imposed on the data model for both Eq. (3.2) and (3.3) [28], include

- $\big\|x\big\| \gg \big\|\Delta x\big\|$,
- $E\big\{x\big\} = E\big\{\Delta x\big\} = 0$,
- and $\mathbf{A}E\big\{\mathbf{s}\big\} = E\big\{\psi(\mathbf{s})\big\} = 0$.

where $\big\|\cdot\big\|$ and $E\{\cdot\}$ are the norm of a vector and the expectation operator, respectively. Following the discussion in [32] [8], ensuring that the loss of information is insignificant, the assumption $\|x\| > \|\Delta x\|$ is imposed on the data realization. For some realizations, the more restrictive assumption $\|x\| \gg \|\Delta x\|$ is considered [56]. Moreover, the assumption $E\{\mathbf{s}\} = 0$ does not represent a restriction of generality [28].

One can consider Eq.(3.2) as a function that explains a linear relationship between $\mathbf{s}$ and significant information in x through the use of a model plane that is defined by the column space of $\mathbf{A}$. On the other hand, Eq.(3.3) is considered as an extension of Eq. (3.2) in a nonlinear sense, where the nonlinear transformation of $\mathbf{s}$ explains significant information in x. Then the objective is to estimate the dimension of $\mathbf{s}$ and determine the significant information.

Several approaches have been proposed for the linear structure. Most of them are related to the application of the PCA method by estimating the column space of $\mathbf{A}$, and rely on various assumptions.

It is important to note, however, that a consistent estimation of d is only guaranteed under the assumption that $E\{\mathbf{s}\Delta x^T\} = 0$ [28]. In contrast to the well-established techniques to estimate d for Eq.(3.2), the research community has devoted comparatively little attention to estimating d in Eq.(3.3) [28]. Global ID estimation methods, such as projection techniques, tend to produce an explicit model surface and/or a reduced set of source signals. In contrast, non–parametric methods, such as fractal methods, generally only provide the ID estimate by itself, without recovering the source signal. The term ‘fractal’ is used since, under this sort of approach, the intrinsic dimensionality d does not need to be an integer. Next, the following sections briefly discuss the techniques for each ID method.

3.3 Local methods

In this section the local intrinsic dimension methods are covered. These methods attempt to estimate the intrinsic dimension by analyzing subsets of the data set. Camastra [8] defined the local methods as the methods that try to estimate the topological dimension of the data manifold where the topological dimension produces a lower bound of ID [8]. Several methods have been proposed to estimate ID locally such as Near neighbor algorithm and Charting a manifold. It is essential to identify a suitable number of subsets (samples) with a small size which ideally lie on the same manifold [54]. In Subsection 3.3.1 the explanation of Fukunaga-Olsen's algorithm is briefly presented. The Near neighbor algorithm and Topology representing network based method are illustrated in Subsection 3.3.2 and Subsection 3.3.3, respectively. In Subsection 3.3.4 the explanation of Charting a manifold method is outlined. Subsection 3.3.5 presents the maximum likelihood estimation method. An implementation of one of the local methods on artificial data sets is presented in Subsection 3.3.6.

3.3.1 Fukunaga-Olsen's algorithm

This algorithm is proposed by Fukunaga and Olsen [72] as the basic algorithm to use to obtain a topological dimension. The feature of the algorithm is the linearization of functions in local regions [32]. The intrinsic dimensionality of the data is obtained by finding the number of random variables d from observed samples.

The algorithm assumes that the data vectors are embedded locally in linear space [8]. In this technique the data set is divided into small regions, which construct linear variable relationships in each region. Practically, it is important to ensure that there are adequate data vectors in each local region. It is also important to note that the estimated dimensionality is too large in the local regions for a limited data set. This is due to that a local region with sufficient points is too large for the surface convolutions at that point [32].

Fukunaga-Olsen's algorithm has the ability to vary the size of the local regions. Fukunaga stated that this variability is critical as the practical problem to obtain the dimensionality depends on the size and number of samples in the local regions [32]. Next, the ID is derived by computing the number of normalized eigenvalues of the covariance matrix which are greater than a threshold [8]. Practically, the eigenvalues are normalized by dividing them by the largest eigenvalue [8]. The drawback of the algorithm is that its computation is complicated [72] and the value of the threshold has to be fixed heuristically.

3.3.2 The Near Neighbor Algorithm

Trunk (1976) used near neighbor techniques to estimate the ID [8] [72]. This algorithm attempts to identify k nearest neighbors for each pattern in the recorded data set, where k is an integer value, and then for each pattern it constructs the subspace which contains data vectors from ith pattern to its k nearest neighbors [8]. The angle is computed between the subspace of ith pattern and the $(k+1)$th near neighbor for all i [8]. The ID estimation is equal to k if the average of these angles is less than a threshold. Otherwise k is increased by 1 and the process is replicated [8] [72]. The drawback of this method is that the choice of the threshold is not quite clear [8].

Pettis et al. [72] improved the technique based on density estimation by assuming that the data has a locally uniform distribution. This technique depends on some factors such as the number of patterns and the maximum value of near neighbors used [72]. The ID is obtained [8] as

$$\mathrm{ID} = \frac{\mu_k}{(\mu_{k+1} - \mu_k)\ k}, \tag{3.4}$$

where μ_k is the mean of the distances from each pattern to its k nearest neighbors. The ID estimate looks biased when this is done [8]. Another algorithm has been proposed by Verveer and Duin [89], which provides a non-iterative solution for ID estimation by fitting a regression line to μ_k as a function of $(\mu_{k+1} - \mu_k)\ k$ in case of observing μ_k for $k = k_m$ to $k = k_M$ [8]. The values $k = k_m$ and $k = k_M$ should be small. Both Pettis' and Verveer and Duin's algorithms are influenced by outliers which tend to affect ID estimation significantly [8], and are also affected by the edge effect [8]. This means that the data points which lie close to the cluster boundary are not uniformly distributed [8]. To overcome this problem, the user needs to eliminate those boundary points and select $k_m > 1$ [8].

3.3.3 TRN-based methods

Martinetz and Schulten (1994) [66] [8] propose the topology representing network (TRN) which is an unsupervised neural network. The algorithm preserves the original topology of the data in the map. The idea is to use Hebbian adaptation rule to form Delaunay triangulation to construct a comprehensive topology representing network [66].

Several papers use TRN techniques to improve other techniques. Bruske and Sommer [8] improved Fukunaga-Olsen's algorithm using TRN. Bruske and Sommer's algorithm performs Voronoi tessellations of the data space, and determines a PCA in each Voronoi set. The method has some limitations. It is necessary to use heuristic thresholds to state the significance of the eigenvalue [8]. Frisone et al. [8] used the TRN method to obtain an ID estimate directly. The ID of a data set is determined as the number n of cross-correlations learnt by each neuron of the TRN. He suggested that, in the Sphere Packing Problem (SPP), the number of n cross-correlations is approximately equal to the number k of spheres which touch a given sphere [8]. Frisones algorithm is limited since the number k is needed to be measured. This is difficult because k is only known for few dimensions. In addition, the number k increases exponentially as the dimensions increase [8].

3.3.4 Charting a manifold

Charting a manifold is a new generation of nonlinear intrinsic dimension estimation methods proposed by Brand [6], which considers the noise around the manifold. The technique assumes that the data lies on or close to a low-dimensional manifold embedded in the high-dimensional space, and that a 1-to-1 nonlinear transformation is mapped between the high dimensional data space and the manifold (vector space) [90] [6].

The basic idea is as follows [6]. Suppose a data set Z where the data points are sampled from a manifold M with the intrinsic dimensionality d where $d \leq D$. The mapping to $\mathbb{R}^d$ should provide a smooth curve which guarantees that the mapping from M to $\mathbb{R}^d$ is linear in some neighborhoods on the manifold [6]. Hence, assume a circle of radius r, placed somewhere in the center of the data cloud, contains $N(r)$ data points. Brand argues that [6], if the underlying manifold is sufficiently smooth, there will be a scale r at which the the manifold is *locally* approximately linear. At the local linear scale, $N(r)$ grows $\propto r^d$, while at noise level, the number of points $N(r)$ will grow $\propto r^D$. We may refer to the former radius, say r_0, as the signal level, at which the points are distributed only in the directions of the local tangent space of the manifold.

Increasing the radius further, the curvature becomes visible so that $N(r)$ will increase at a rate between r^d and r^D. When reaching the boundary that encloses all data, $N(r)$ eventually flattens. Brand's expression is

$$G(r) = \frac{\partial \log \big(r\big)}{\partial \log \Big(N\big(r\big)\Big)} \tag{3.5}$$

that determines the radius r_0 to derive the intrinsic structure best, and ∂ is a derivative symbol. Hence, according to above considerations [6]:

- at noise scales $G(r) \approx \frac{1}{D} < \frac{1}{d}$,
- at the scale where the curvature becomes significant $G(r) < \frac{1}{d}$.
- at the locally linear scale, the process peaks at $G(r)$, with maximum

$$G(r_0) = 1/d.$$

Hence, one can read the intrinsic (topological) dimension d directly from the graph $(r, G(r))$. Although this concept is appealing in practice, its implementation is nontrivial.

Since it is a local method, the technique needs to be repeated over several target points (corresponding to the centers of the r-balls), and the resulting local IDs need to be averaged. The choice of target point is important, since the topological dimension at the boundaries is smaller than that of the manifold itself. We discuss the choice of target points in Chapter 4 and show how the ID can be obtained from the log-log plot using nonparametric or parametric regression approaches.

3.3.5 Maximum Likelihood Estimation

Maximum Likelihood estimation method (MLE) was proposed by Levina and Bickel [60] to obtain the intrinsic dimension of a data set. Levina and Bickel also studied the statistical properties of the estimator. This technique assumes that the observations are independent, and it applies the principle of maximum likelihood to the distances between close neighbors [60]. As for the k-NN algorithm, those neighbors lie on the same manifold [54]. The observations in the ball are treated as a homogeneous Poisson process and the ID estimate is derived by maximizing the log-likelihood function. The dimensionality is estimated by computing the number of neighbors contained in a sphere [60]. The sphere is assumed to be small enough and to contain enough data points.

Similarly as in charting a manifold, suppose a sphere of radius r is around a fixed point x. The ML estimator works as follows [60]. Let k be the number of nearest neighbors to the point x_i. Then, for fixed k, define the quantity

$$d_k\big(x_i\big) = \left[\frac{1}{k-1}\sum_{j=1}^{k-1}\log\left(\frac{T_k\big(x_i\big)}{T_j\big(x_i\big)}\right)\right]^{-1}, \tag{3.6}$$

where $T_k\big(x_i\big)$ and $T_j\big(x_i\big)$ are the Euclidean distance between x_i and the kth and jth nearest neighboring samples, respectively. One can divide by $(k-2)$ instead of $(k-1)$ to obtain an asymptotic unbiased estimator [60]. The method assumes that all the data points come from the same manifold, and therefore average over all observations [60]. Now the ID is obtained locally at every data point by computing the average dimension estimation within the data sphere as:

$$d_k = \frac{1}{N}\sum_{i=1}^{N} d_k(x_i),$$

The process is repeated for each value of k within the range. Finally, the intrinsic dimension for a data set Z can be obtained by averaging over a range of k:

$$d\,(Z) = \frac{1}{k_2-k_1+1}\sum_{k=k_1}^{k_2} d_k. \tag{3.7}$$

The method produces satisfactory results on a range of simulated and real data sets [60]. The drawback of this method is that the estimator suffers from a negative bias for large values of k [60]. This bias decreases with the growing of sample size [60]. On the other hand the bias increases with high dimension because it needs a very large data sample in the sphere [60]. Furthermore, the negative bias could be caused by edge effects.

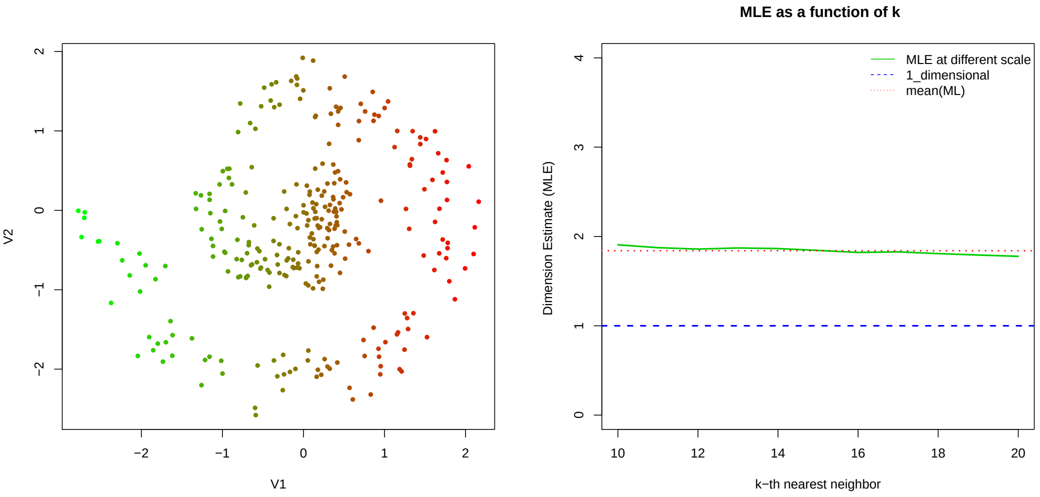
Mackay and Ghahramani [63] discussed the bias in the estimated dimension and suggested a bias correction of MLE by averaging the inverse of the estimator. Adapting to Levina and Bickel's work, we propose to replace Eq.(3.7) by the median, and illustrate the performance of this technique in Chapter 4.

3.3.6 Experiments of local method on artificial data sets

In order to evaluate the performance of a local method, the MLE method is implemented in the R software [74] and applied to two artificial data sets: Spiral data and Swissroll data. The data set is scaled to mean 0 and variance 1. Note that when applying MLE to a data set, the choice of the parameter k is very important, where k is the selection of the number of nearest neighbors. Practically, for small numbers of neighbors k, the MLE algorithm provides an unreasonable value of dimension estimation. This leads one to infer that the algorithm has not yet worked. Furthermore, the intrinsic dimension estimation is frequently low when k increases. We use a reasonable range of k between 10 and 20 as advised by Levina and Bickel [60]. In addition, we test the sample size effect on the MLE method by computing the dimensionality at several sample sizes. In practice, for simplicity and computation time, a maximum sample size value of 300 data points is taken.

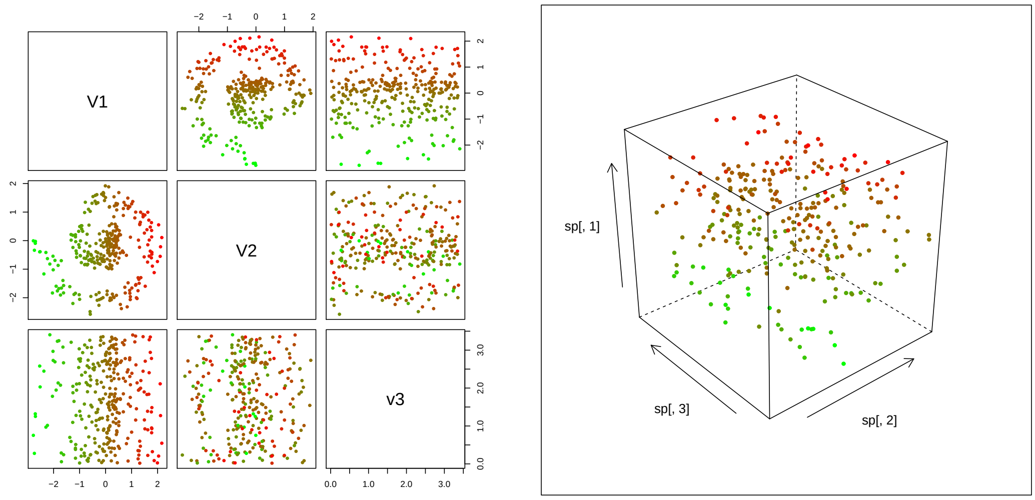
MLE applied to Spiral data and Swissroll data

The Spiral data consists of points randomly sampled from a one-dimensional nonlinear manifold embedded in a two-dimensional space. The data consists of 300 data points, as displayed in Figure 3.1a, which illustrates that the intrinsic dimensionality of data is equal 1. On the other hand, the Swissroll data consists of three variables with 300 data points. It is generated by adding the uniform variable to a Spiral data, as displayed in Figure 3.2b, which delineates that the intrinsic dimensionality of this data equals 2.



(a) (b)

Figure 3.1: (a) A 2D scatter plot of scaled Spiral data, (b) The dimensionality estimation of scaled Spiral data via maximum likelihood estimation with 300 data points.



(a) (b)

Figure 3.2: (a) A scatter plot matrix of scaled Swissroll data, (b) A 3D scatter plot of scaled Swissroll data.

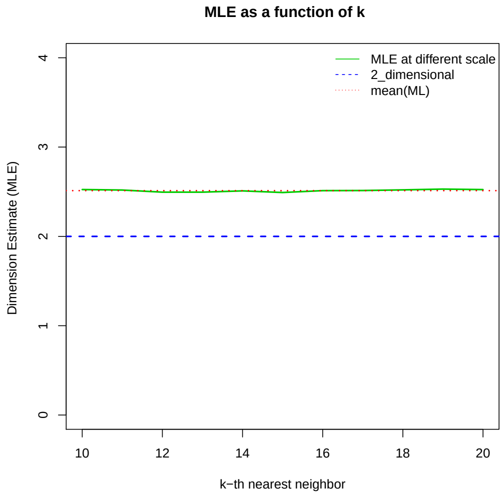


Figure 3.3: The ID estimation of scaled Swissroll data via maximum likelihood estimation with 300 data points.

			Sample size				
Data set	D	True (ID)	50	100	150	200	300
Spiral	2	1	1.73	1.71	1.80	1.73	1.84
Swissroll	3	2	2.87	2.69	2.52	2.47	2.51

Table 3.1: The MLE estimate for artificial data sets in different sample sizes.

In Table 3.1, the ID estimates are obtained via the MLE method using different sample sizes. From table 3.1 we observe that the performance of MLE is influenced by the sample size and the parameter k. Besides the computation time of implementation increases when increasing the sample size. The resulting estimate is depicted in Figure 3.3, which shows different estimations over the range of k of Swissroll data with sample size 300, and the final estimator is 2.51. Levina and Bickel [60] observe that, for dimension estimates equal to 2, the required sample size has to be 1000 to obtain an estimate near to the true value (In this context, 300 is small sample size). We observe that MLE method gives a visual impression of positive bias but is consistent with the scree-plot (linear PCA). Further more, the computation time of implementation increases exponentially as the sample size increase.

3.4 Global Methods

The majority of methods used for estimating the ID depend on global techniques, such as PCA or maximum likelihood PCA [87] [47] [56]. These methods try to estimate the ID by studying the structure of the entire data set. Global methods try to estimate the dimension using the whole data set [8], and imposing the implicit assumption that the intrinsic dimension is constant over the data set. The core concept is to unfold or flatten the data in a high-dimensional space. Global methods can be grouped [8] into projection techniques, multidimensional scaling methods and fractal-based methods. In Subsection 3.4.1 the projection techniques are illustrated, by the example of linear and nonlinear PCA methods. Multidimensional scaling methods are briefly presented in Subsection 3.4.2. Fractal-based methods and their estimation methods are discussed in Subsection 3.4.3. An implementation of one of the global methods on artificial data sets is discussed in Subsection 3.5.2.

3.4.1 Projection techniques

Projection or eigenvalue techniques are based on PCA techniques, where PCA projects the data points onto lines or planes spanned along the direction of maximal variance. Then one computes the eigenvalues and the eigenvectors of the covariance matrix of the recorded data. These methods can be divided into linear and nonlinear methods, as previously explained in Chapter 2.

A. Linear PCA

Linear PCA is a simple transformation carried out in order to minimize the mean square reconstruction error. The ID is obtained as the number of eigenvalues of Σ greater than a given threshold [60]. Several approaches determine the number of (retained) components derived by PCA, such as cross–validation, the scree plot and the broken–stick model. Some of the stopping rules are briefly illustrated below.

Stopping rules for linear PCA method

Jackson (2003) [46] presented a survey on several stopping rules in PCA analysis and provided a comparison between those rules. The objective of stopping rules is to determine the number of principal components that should be retained. Those approaches are cross–validation, the scree plot, the broken–stick model and the proportion of total variance. Other approaches are included in Jackson [46] and Kruger et al. [56].

a. Scree plot - this is the plot of each eigenvalue λ_j against the component index j in descending order. Cattell [56] illustrated that the scree plot displays two sets. The first set is of the first few eigenvalues that decrease sharply, while the second set is the remaining eigenvalues which decreases slowly. Then the retained eigenvalues are the first set which includes the first eigenvalue of the second set [46] [56]. The drawback of this method is that it often overestimates the number of components that are retained [46].

Another way to detect the retained components is based on visual impression by determining the knee of the scree plot, which is done by eye.

b. Broken-stick - this method is proposed by Frontier and based on the eigenvalues from random data. The model assumes that the eigenvalues distribution follows a broken-stick distribution when the total variance is divided randomly amongst the different components [46]. Therefore, the significant eigenvalues are those that override the generating eigenvalues via the broken-stick model, where the generating eigenvalues could be computed as [46]

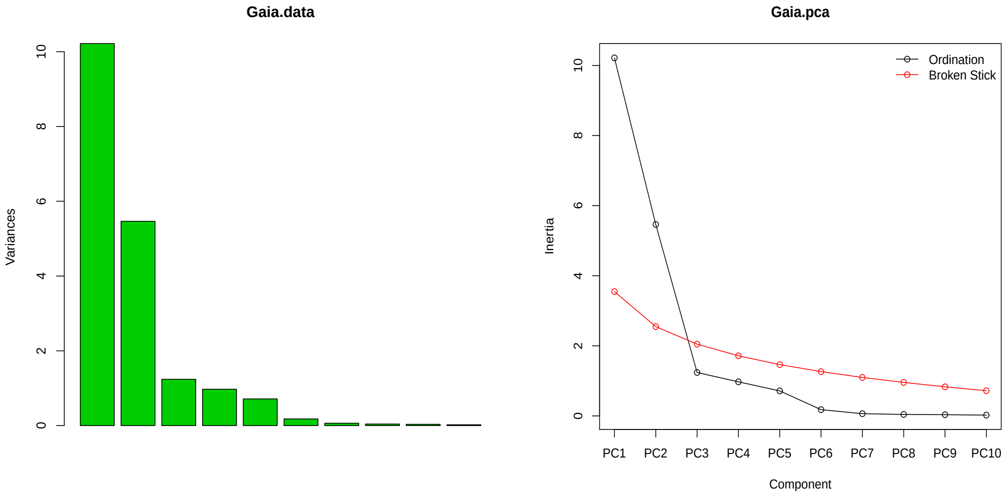
$$\tau_k = \sum_{i=k}^{D} \frac{1}{i},$$

where the number of variables is denoted as D and λ_k is the size of eigenvalues for the k^{th} component under the broken-stick model. Compared to other statistical approaches, this method presents an accurate dimensionality estimation [46].

c. Proportion of total variance - in general the sum of the variances of the data variables is equal to the sum of the eigenvalues of the data covariance matrix. One can decide the portion of total variance to be preserved, then the retained principal component included all the components up to some proportion of total variance [46]. If one chooses a threshold, for example 95% or 99%, then the number of components can be selected that exceed this threshold. Although this method is simple to implement, the selection of the threshold is arbitrary and could lead to an underestimation of ID [46].

Figure 3.4a displays an example of the scree plot approach on the Gaia data [25] with 19 variables, the scree plot shows that three components explain 89% of the total variance of the scaled data, while four components explain 94% of the total variance. This example will be discussed in details in Chapter 5. In contrast Figure 3.4b shows an example of broken-stick method on Gaia data, it shows the first few eigenvalues fall sharply while the smallest eigenvalues tend to lie along a straight line (black line).

On the other hand, the linear PCA method that is based on linear approximation and its stopping rules [46] [56] fail for nonlinear manifolds.



(a) (b)

Figure 3.4: Gaia data; (a) eigenvalue λ_j against the principal component index j, (b) the black line represents eigenvalue λ_j against the component index when applying PCA on original data, the red line represents eigenvalue λ_j against the component index when applying PCA on randomly generated data .

B. Non–Linear PCA

Nonlinear PCA methods have been suggested to solve the limitations of PCA. There are three approaches of nonlinear PCA: principal curve, autoassociative neural network and kernel PCA. Principal Curve (PC) is a smooth one-dimensional curve passing through the middle of a data cloud. The concept of principal curve assumes that the intrinsic middle structure of data is a curve rather than a straight line. An autoassociative neural network (ANN) is determined by means of a five-layer neural network. The layers are: input layer, mapping layer, bottleneck layer, demapping layer and output layer. The ID is determined from the number of the neurons in the bottleneck layer [8]. Although this method performs better than PCA, it has some limitations. The projections are suboptimal and unsuccessful when curves or surfaces intersect themselves [8]. The Kernel PCA approach maps the data $Z \in \mathbb{R}^D$ into a high-dimensional feature space $Z \mapsto \Phi(Z)$. Then the principal component analysis is performed on $\Phi(Z)$. The method makes a nonlinear projection of the data set in a new space. Then the eigenvalues of the covariance matrix are calculated. Therefore, the ID is obtained as the number of non-null eigenvalues [8]. The Kernel PCA technique is influenced by the kernel choice, and due to noise, the last eigenvalues are not null. Therefore, similar as for linear PCA, it is better to neglect the eigenvalues whose magnitude is lower than a threshold value [8].

For these techniques various approaches to determine d have been considered, including cross-validation [82], an analysis of the residual variance [56] and the H principle [43].

3.4.2 Multidimensional scaling method

Multidimensional Scaling (MDS) is a nonlinear projection technique. The technique attempts to project the data set in such a way that preserves the pairwise distances between data points [8]. A brief review on MDS algorithms is presented in Chapter 2. Now consider Kruskal stress as explained in Section 2.3.3. The dimensionality is obtained by plotting the minimum stress against the dimensionality of new (output) space. Therefore the ID is the value for which there is a knee or a flattening of the curve [8]. The drawback of this algorithm is that in some cases the knee does not exist [8]. Camastra explained that with Bennett's algorithm, see Section 2.3.3, the patterns in the input space are moved to increase the variance of the interpoint distances. Then adjust the position of the patterns which make the rank of interpoint distances the same in all local regions. The process is iterated until the variance of the interpoint distances levels off [8]. The covariance matrix is computed by the previous steps. Therefore, the ID is derived as the number of significant eigenvalues of the covariance matrix [8].

3.4.3 Fractal-based methods

In this Section we introduce the concept of fractal dimension. Fractal is a term for the geometrical structure of an item, with self-similarity and symmetry properties which imply that the original data structure can be divided into substructures with the same form at any selected scale [65]. To put the analogy into a statistical perspective: while fractals can be considered as mathematical *sets* with non–integer dimension, in fractal dimension estimation we deal with *data sets* of non–integer intrinsic dimension.

As illustration, the Koch curve can be divided into small copies of itself, the number of copies $N = 4$ with scaling factor $r = \frac{1}{3}$, displayed in Figure 3.5. Then the intrinsic dimension of the curve is

$$d = \frac{\log(4^N)}{\log(3^N)} = \frac{N\log(4)}{N\log(3)} \approx 1.2619,$$

and one can infer that the curve is expected to be more than a line and less than a plane. Practically, large values of fractal dimensions indicate that the objects are roughly irregular whilst small values indicate that the objects are smooth [7]. Fractal applications are widely used in many natural applications such as snow accumulation in forests [73], tree crowns [96], recognition of computer vision [13] [71], chaos theory [85] and in time series analysis [18]. Although fractal dimension methods are useful, many literature have found that sometimes it is difficult to explain the different (biased) results that are provided by the dimension estimators [7].

Fractal dimension is a measure that describes the geometry of an irregular object

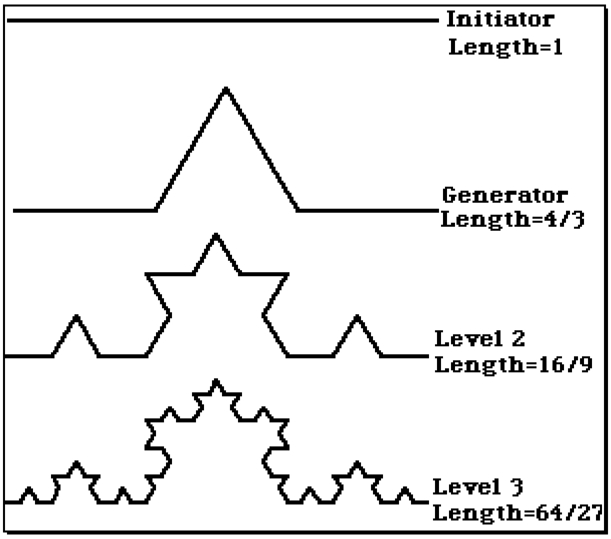


Figure 3.5: Koch curve construction (taken from [65]), there are 4^N line segments with length $\frac{1}{3^N}$ and for $N \to \infty$ then the fraction $(\frac{4}{3})^N \to \infty$.

(here: a data set) by an estimated real number. It describes the filling of the fractal object's space, which can be used to construct ID estimators. Various fractal-based methods have been proposed, including quantization estimator [75], kernel correlation [41] method, horizontal structuring element, box-counting and correlation dimension [91] [96] [69]. Camastra surveyed intrinsic dimension methods with focus on fractal-based methods [8] [9]. Box-counting and correlation dimension methods are most commonly used and provide non-linear methods.

It is noted that the Hausdorff dimension is bounded above by the box-counting dimension. The box-counting dimension is preferable in practical applications because it is easier to evaluate [78] [8] [9].

Box-counting dimension - the approach is also referred to as the capacity dimension of a data set [85]. It is the more popular with scientists because of its simplicity and because it requires less computational time. The idea is as follows. For any bounded subset Z of $\mathbb{R}^D$, partition the embedding space $\mathbb{R}^D$ into a grid of boxes of side-length r, where each box contains at least one data point. Let $N(r)$ be the number of boxes that are required to cover the object's space with r being the box size. Then the box-counting fractal dimension is defined as

$$d_{box} = \lim_{r \to 0} \frac{\log(N(r))}{\log(\frac{1}{r})} = -\lim_{r \to 0} \frac{\log(N(r))}{\log(r)}, \tag{3.8}$$

where the negative sign is necessary as the numerator is positive and the denominator is negative. Obviously the number of boxes $N(r)$ increases proportional to the scale r, i.e. $N(r) \propto r^{d_{box}}$. In practice, the fractal dimension is determined by using a loglog plot where a curve of $\log(N(r))$ is plotted versus $\log(r)$. Then the dimension is estimated as the slope of the linear part of the curve [8].

Although the algorithm is easy to use there are some drawbacks. All boxes should be the same size which could lead to an empty box. Furthermore it increases the time of computation since the program has to determine the nonempty boxes for each data point [85]. Therefore the technique's complexity will increase exponentially with the dimensionality of data set. More generally, for those reasons, Box-counting dimension can be computed for low-dimensional embedding space [8].

Correlation dimension - this is commonly used to estimate the fractal dimension. The idea of the correlation dimension method is to estimate the intrinsic dimension via a pairwise distances algorithm which counts the number of point pairs that are closer to each other than a given radius. Grassberger and Procaccia [35] introduced the correlation integral algorithm, named the GP method, which is used to define the correlation dimension estimation from a given data set. Now the correlation integral, according to GP method [9], is defined as the proportion of distance points less than r, that is

$$C(r) = \lim_{N \to \infty} \frac{2}{N(N-1)} \sum_{i=1}^{N} \sum_{j=i+1}^{N} I\left(\|x_j - x_i\| \leq r\right), \tag{3.9}$$

where $I(.)$ is an indicator function, and $\|x_j - x_i\|$ denotes the Euclidean distance between data points, x_j and x_i. Note also that the number of data pairs which can be formed from N points is given by $\begin{pmatrix} N \\ 2 \end{pmatrix} = \frac{1}{2}\, N(N-1)$, which is just the inverse normalizing constant, so that clearly $0 \leq C(r) \leq 1$. Now Let

$$D(r) = \frac{\log C(r)}{\log(r)},$$

then the correlation dimension is defined by:

$$d_{cor} = \lim_{r \to 0} D(r). \tag{3.10}$$

Therefore, for small r, the dimensionality can be obtained as the slope of the (linear part of) the 'loglog' curve of $\log(C(r))$ versus $\log(r)$ [8].

In addition, although the method is simple it has drawbacks. Some papers discuss the challenges that arise with box-counting and correlation dimension methods [93] [69].

Theiler [85] outlined the following:

- For very small r, meaning that the circle contains few data points, the number of pairs inside the sphere is influenced quickly by the noise. In addition, one could get a negative slope in the loglog plot.

- An accurate dimension estimation requires large N and it is difficult to deal with large N since we consider the error of estimate.

- The dependency of $C(r)$, i.e. $C(r + \Delta r)$ is dependent on $C(r)$.

- The error in the estimation can not be computed from the loglog plot.

It is worth mentioning that some relevant literature has underestimated those problems in view of the ease of implementation. Grassberger at al. present improvements to the correlation integral $C(r)$ which tackle some of those issues [93]. Several techniques have been proposed to compute an optimal estimate of the correlation dimension. Taken's method [83] used the Fisher's maximum likelihood rule to obtain the correlation dimension with minimal standard error. He used a finite set of distances pairs and presented the way to choose the scale radius. In addition, when drawing a loglog plot of $C(r)$ and r, one notices that the curve at the upper end, when r increases to a certain value, bends down and becomes a plateau and $C(r)$ approaches 1 [69].

Generally, the fractal dimension of a data set is affected by several factors: the relationship among variables, data dimensionality, the intrinsic dimension of the data set, the portion of distance pairs that are used for calculation, and the sample size N [69]. Notably the definition (3.9) of the correlation integral would require an infinitely sized data set. In order to arrive at an accurate dimension estimation the number of data points needed is estimated as $N = 10^{D/2}$ [8]. Compared to the box–counting dimension, the correlation dimension is in practice less demanding about the sample size, and has a larger dynamical range of $O(N^2)$. Furthermore, it can be evaluated for smaller values of r [85] [35].

The main problem with the practical implementation of the correlation dimension is that the correlation integral needs to be estimated for a ball of radius tending to 0. Clearly, the radius r can not be equal to zero because this implies that there are no data points in the circle, yielding "NAN" at $C(0)$. Hence, one needs to decide on a suitable range of values of r which is used to arrive at an estimate of the ID [85].

With our techniques we try to capture the distance pairs of $C(r)$ in a more effective way which is consistent with the GP method. The algorithms achieve the estimation of the ID of a given data set at radius $r = 0$. The improved methods are described in the following Chapter.

3.5 Remarks on global methods

3.5.1 Justification of correlation integral

It is important to reflect why (3.10) is a sensible expression to define. To this end, consider a structure with lies (perfectly) on some (linear or nonlinear) subspace of Z. Then it is easy to see (we discuss this later in Remark 2 below) that $C(r) \propto r^d$ for sufficiently small r. In other words, one has

$$C(r) = c \cdot r^d,$$

where d is the intrinsic dimension and c is constant. Now, applying the logarithm to the above equality, we get

$$\log(C(r)) = \log(c) + d \log(r).$$

By substituting into equation (3.10), one finds

$$\begin{aligned} d_{cor} &= \lim_{r \to 0} \frac{\log(C(r))}{\log(r)} \\ &= \lim_{r \to 0} \frac{\log(c) + d \log(r)}{\log(r)} \\ &= \lim_{r \to 0} \frac{\log(c)}{\log(r)} + d \frac{\log(r)}{\log(r)} = d, \end{aligned} \tag{3.11}$$

that is, the correlation dimension indeed recovers the intrinsic dimension of the data set [28].

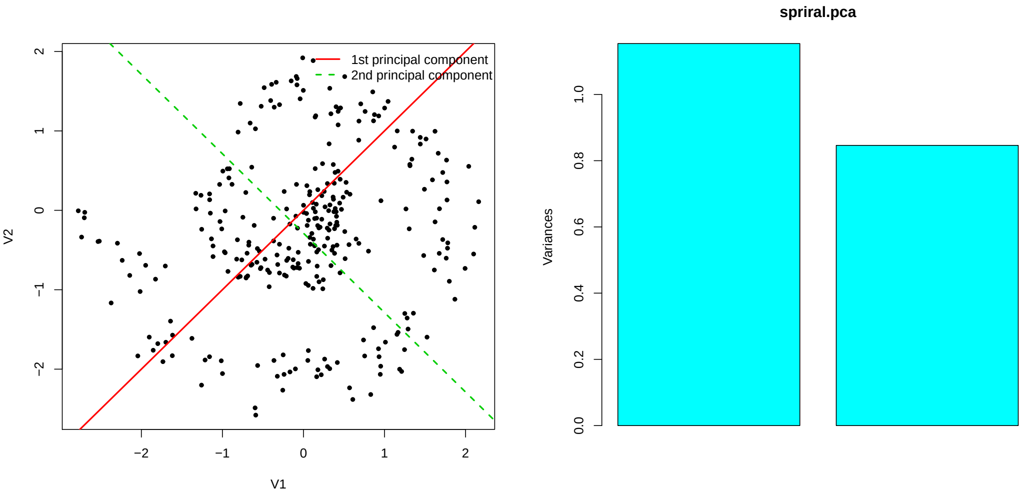
Further, we need to justify why, for data of intrinsic dimension d, one should expect $C(r) \propto r^d$. With Subsection 3.3.4 in mind, this may appear counter–intuitive, since one may feel that, if the number of points within the r–ball increases with r^d, then the number of pairs should increase with order $O((r^d)^2) \propto r^{2d}$. This apparent contradiction is resolved by realizing that in Subsection 3.3.4 we deal with a *local* method, where the $r-$ ball is successively expanded starting from some target point on the manifold, while, under the scenario of this subsection, we are not tied to a target point, but count pairs *globally.* To make this plain, consider a simple scenario with N data points sitting at discrete positions (with distance 1) along a line:

$$\bullet \qquad \bullet \qquad \bullet \qquad \bullet \qquad \cdots \qquad \bullet \qquad \bullet$$

Then, for $r = 0$, the double sum in the numerator of (3.9) is 0. For $r = 1$, this sum is $N - 1$, and for $r = 2$, it is $(N - 1) + (N - 2)$. Eventually, for general r, this sum is $(N - 1) + (N - 2) + \cdots + (N - r) \approx Nr \propto r$ for large N, confirming the alleged statement in the case $d = 1$ [28]. For non-linear structures, this statement would still hold for sufficiently small r.

3.5.2 Experiments of global methods on artificial data sets

In this section, the implementation of linear PCA, nonlinear PCA (Kernel PCA) and Multidimensional Scaling (MDS) are provided to determine the intrinsic dimension as global methods. In addition, we provide an implementation of the LLE method on the data sets. We mentioned in Subsection 3.3.6 that the intrinsic dimension of Spiral data is equal to 1 while for Swissroll data the intrinsic dimension equals 2. The linear PCA, Kernel PCA , LLE and MDS methods are implemented in software



(a)

(b)

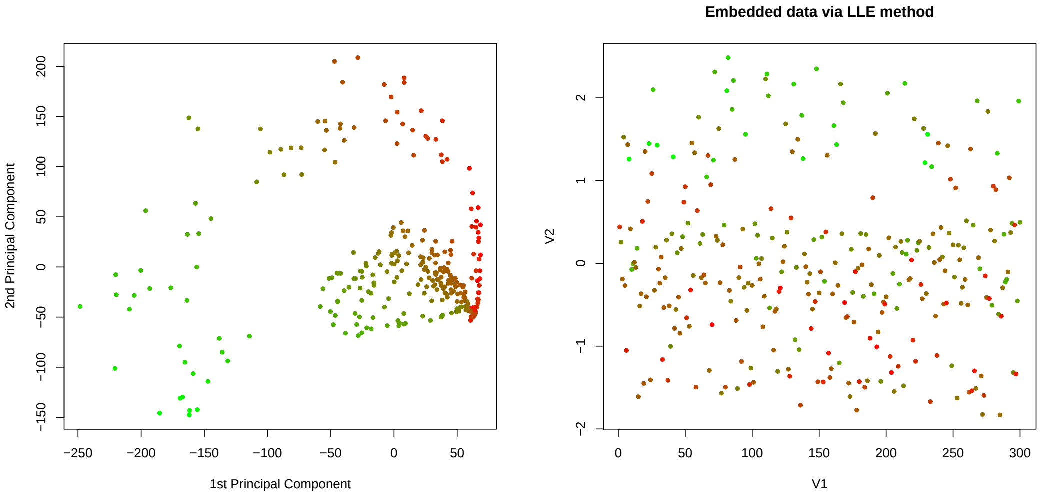
Figure 3.6: Spiral data: (a) Principal Components Graph, (b) Scree plot of linear PCA from scaled Spiral data.

R [74]. More precisely, the code of Kernel PCA is available in the 'kernlab' Package, the MDS code is available in the 'MASS' Package and the code for the LLE method is found in the 'lle' Package. The methods are applied to two artificial data sets: Spiral data and Swissroll data, where both data sets are scaled with mean 0 and variance 1.

Firstly, the linear PCA is implemented on the Spiral and Swissroll data sets. The results are displayed in Figure 3.6 and Figure 3.8, respectively. Figure 3.6a illustrates the first two components for the Spiral data which explain 58% and 42% of the total variance, as shown in Figure 3.6b. One can conclude that the (linear) ID for this data set is 2. The ratio between the eigenvalues of the components is equal to 1.36.

For the Swissroll data, Figure 3.8a shows that two principal components explain 69% of the total variance. Consequently, one can conclude that the (linear) ID of the Swissroll data set is 3.

Secondly, the application of Kernel PCA on the Spiral data is discussed. Figure 3.7a shows the output after Kernel PCA is applied. We use the polynomial ker-



(a) (b)

Figure 3.7: Spiral data: (a) The dimensionality via Kernel PCA method, (b) The output after applied LLE method.

nel function with degree 2 and scale 2. The ratio between the eigenvalues of the components is equal to 1.22, which is less than for the linear PCA. The result illustrates that the two eigenvalues for Kernel PCA method are even more equal than for PCA, which means that KPCA has failed totally to identify the one-dimensional curvilinear substructure in this data. It is important to note that the performance of Kernel PCA is affected by the kernel function and the parameter changes of the function.

In addition, the LLE method is applied to Spiral data. Figure 3.7b shows the output after LLE is applied, which produces nicely following colors from left to right. The result confirms that the LLE method has identified correctly ID =1, which is the true ID.

Now, consider Swissroll data. Practically, Kernel PCA does not provide satisfactory results and problems arise when standard kernel functions are used. It is known that the method performs poorly on the Swissroll manifold. Consequently, the MDS algorithm is used to obtain a $2D$ embedding, using Sammon stress. We used R function **sammon** in Package ‘MASS’ [74]. The result is displayed in Figure

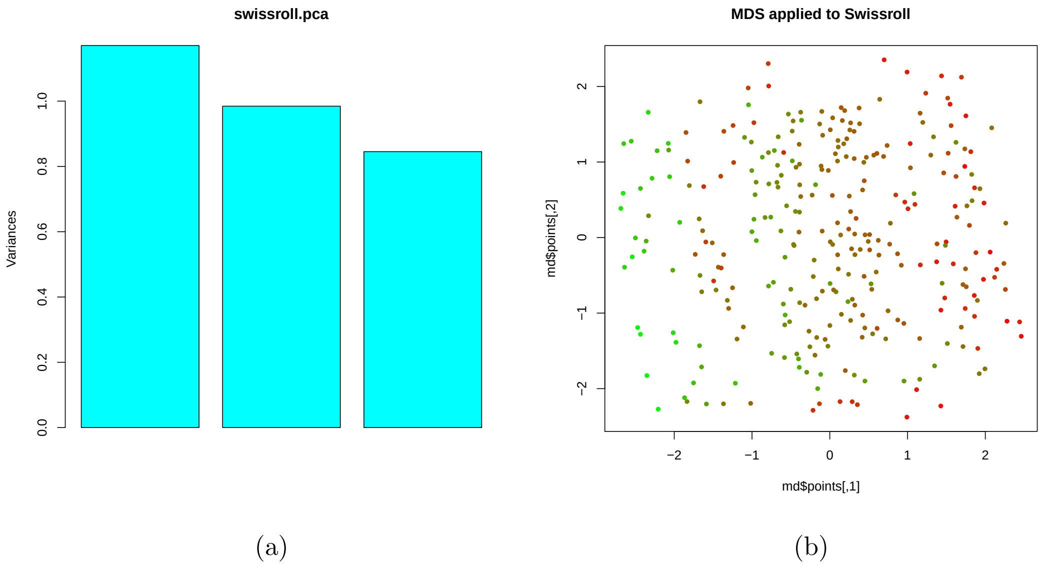
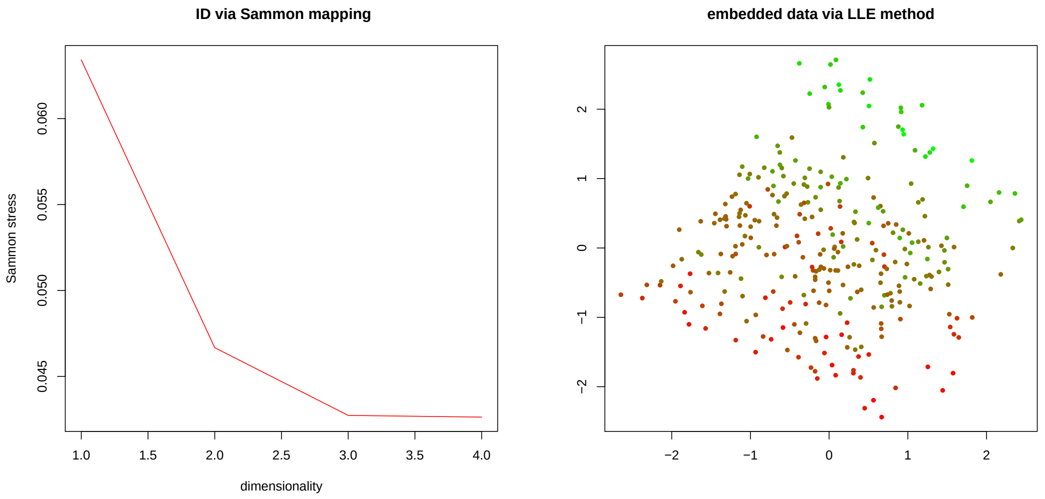


Figure 3.8: Swissroll data: (a) Scree plot of linear PCA from scaled data, (b) The output of reduced data after the MDS method is applied on the scaled data.

3.8b. Now, to obtain the ID, as illustrated in Subsection 3.4.2, plot the minimum stress against the dimensionality of new (output) space, and the ID is the value for which there is a knee, here equal to 2, as shown in Figure 3.9a. In addition, Figure 3.9b shows the embedding result of the LLE algorithm. We can observe that LLE unrolls the $3D$ data set into a plane. We observe that techniques, such as PCA and Nonlinear PCA that do not employ neighborhood graphs, provide unreasonable results on these data sets, and that the MLE method, as shown in Subsection 3.3.6, provides an overestimated ID. In addition, the methods implemented previously are basically used as dimension reduction methods.

To summarize, the bias in the ID results came from different reason. MLE method provide bias, as shown in Subsection 3.3.6, because the neighbors need to contain sufficient data points which is difficult for a finite sample size. On the other hand for PCA the bias appears due to the linearity constraint. For Kernel PCA, the bias comes from the specific nonlinearity constraint imposed, which is influenced by the kernel function and the parameter changes of the function. More discussion will be presented in Chapter 5 and 6.



(a) (b)

Figure 3.9: Swissroll data: (a) The dimensionality via MDS method, (b) The output of reduced data after applied LLE method.

3.6 Conclusion

In this chapter we presented an overview of the intrinsic dimension estimation methods. Figure 3.10 displays a classification of techniques for estimating dimensionality. It represents the distinction between techniques due to global and local methods. It is worth highlighting that while the intrinsic dimension in the left-bottom column of Figure 3.10 provides an integer value, it may be both a real number or an integer in the right-bottom and middle columns. As for fractal methods the non-integer character of dimension is made explicit through the term ‘fractal’.

For local methods, divide the data set into small subregions, or provide a series of local ID estimates at several target points, in order to arrive at a suitably averaged overall ID estimator. In practice the ID methods are influenced by several factors, such as computation time, limited size of the data set and noise. It is necessary to insure that the local region contains enough data points to analyze, and with a limited data set it is possible that the local region is too large, which could lead to an overestimation of the dimensionality. On the other hand, the small local region will decrease the eigenvalues due to the noise point [32].

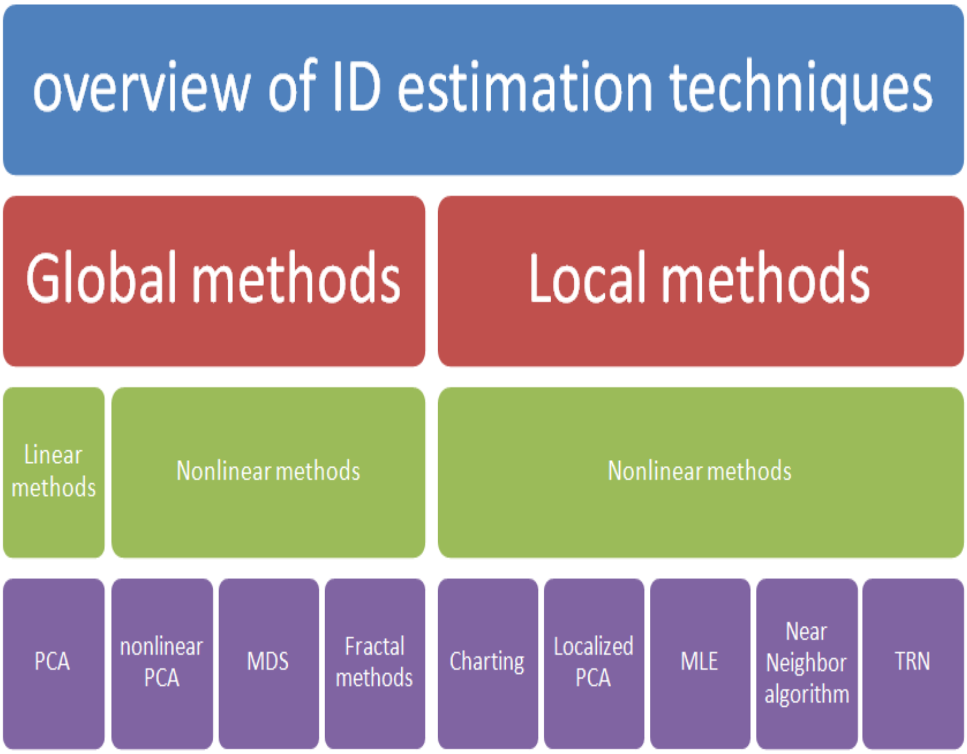


Figure 3.10: Intrinsic dimension methods

Local methods suffer from the presence of the outliers, because the outliers are linked to their k nearest neighbors. To deal with these issues the outliers can be removed before analysis by using the edge points by some criterion, or by using the points that have the highest density, as described later in this thesis. From our experimental results, local methods lead to high computational costs because they determine the dimensionality for each subset. Furthermore, local methods are influenced by the structure of the data (linear, connected two branches). Some methods need to fix the value of the threshold heuristically such as in Fukunaga–Olsen's algorithm, and Bennett's algorithm. Besides the main drawback with the topological dimension is that it is difficult to estimate ID with a finite sample.

Global methods try to estimate the dimension using the whole data set, and imposing the implicit assumption that the intrinsic dimension is constant over the data set. Global methods are extensively used in the manner of projection methods such as PCA, although both these and MDS are dimension reduction methods rather than dimensionality estimation methods. Other methods of dimensionality estimation are indeed only used to estimate, rather than reduce, dimensionality. These include Brand's algorithm, the MLE approach and fractal-based methods.

In general all methods, *local* and *global*, suffer from a negative bias of high dimension, where the bias appears to be due to inadequate sampling. This occurs when the sample is from the region near the edges or boundaries of a manifold [54]. With global methods, these regions provide a too low-dimensional ID estimate and a strong negative bias [54]. On the other hand, the correlation dimension has the smallest bias and the MLE has the next smallest bias [60]. Lastly all methods require large samples in high–dimensions which could increase the computational cost.

Chapter 4

Implementation for Methodology of ID Estimation Methods

In this chapter we introduce new approaches that improve the practical algorithms which determine the estimation of dimensionality whether the underlying data structure is a linear or nonlinear structure, with special consideration for recent developments in non-linear techniques. Our approaches focus on the algorithms based on the concept of charting manifolds (local method) and the correlation-dimension concept (global method), and also deal with their issues that were discussed earlier in this thesis.

4.1 Introduction

For ID estimation, a few approaches exist which, similarly to linear principal component analysis, propose to estimate d, where $d \leq D$, by analyzing the entire data set. In contrast, local methods operate at a specific target point which we denote by x, where $x = (x_1, \ldots, x_D)^T$. However, even for local methods, some researchers state that some sort of averaging over different subregions or target points is essential in order to determine the intrinsic dimension of the full data set [85]. Arguably this averaging step gives local methods a global character as well, though we continue to refer to them as local methods in this presentation. A brief review of the algorithms is provided in Chapter 3.

As far as we know although various nonlinear methods, global or local methods, are available, it seems that not enough work has been done on implementing the methodology of dimensionality estimation of non-linear manifolds. Furthermore, with many methods there is not enough evidence that they work well practically. One example is charting manifold where one needs to select the target points. Furthermore fractal methods require the construction of the correlation integral, from which the ID is extracted using appropriate techniques. This step is not straightforward, since the number of data pairs within a ball of radius tending to 0 need to be counted.

In this chapter we will explore new approaches for computing the estimation of dimensionality. The algorithms can be regarded as nonparametric methods. The techniques will implement some ID estimation methods and obtain the accurate ID. Moreover these approaches address the issues that arise out of counting the number of data points, or numbers of data pairs, which fall within certain balls of given radius r. The new approaches obtain the ID via Brand's charting manifold and via the fractal-based-method, which are nonparametric and nonlinear intrinsic dimension estimation methods. Generally, the nonparametric technique is used if the parametric technique is not sufficiently flexible, and it allows a reduction of the possible modelling biases of parametric techniques. Specifically, the Dip and Regression methods are variants of Brand's algorithm which are considered as local ID methods. The improved methods of correlation dimension, which are the Intercept, the Slope and Polynomial methods, are global ID methods. The localized correlation integral method is an approach that could be defined as a local version of global ID methods. All these techniques provide a reasonable ID estimate when there are a sample of observations or full data set.

The new approaches will be delineated in detail in the next few sections. The improved methods – Dip method and Regression methods – are discussed in Section 4.2. Section 4.3 describes the Intercept methods, Slope method and Polynomial method. Localized correlation integral is explored in Section 4.4. The computation on maximum likelihood estimation is discussed in Section 4.5. An implementation of our approaches on artificial data sets is presented in Subsection 4.6. Section

4.7 presents a brief explanation of software that used in the thesis. Section 4.8 presents a discussion on our approaches that proposed for estimating the intrinsic dimension. The applications of the methods and the results will be outlined in next chapter to demonstrate the working of the approaches. We begin with the practical computation of a local method with charting a manifold.

4.2 Intrinsic dimension via Brand's charting manifold

Brand [6] proposed a concept based on a charting manifold where the intrinsic dimension is obtained by examining the growth rate of samples in hyper-spheres [6]. The algorithm considers the number of data points $N(r)$ who have fallen in certain hyper-spheres. The Subsection 3.3.4 has briefly reviewed concepts of the Brand's algorithm. The technique is implemented using the following steps:

- Step 1: Begin at target points x.
- Step 2: Compute the Euclidean distances between the data points and the selected target point.
- Step 3: Calculate the following equation
$$G(r) = \frac{\log(r)}{\log(N(r))}, \tag{4.1}$$
- Step 4: Sketch the loglog plot.

These steps are demonstrated in detail in the Subsection 4.2.3. In practice, the practical implementation of Brand's algorithm requires the following issues to be considered:

1. The choice of target point x. It is obvious that the more central observations lead to higher ID.
2. The determination of the range of r values.
3. How to deal with the appearance of multiple peaks in the loglog graph.

4. The possibility that the expression $\log(N(r))$ in the denominator could be undefined for small r.

5. How to derive the ID estimation of the entire data set by the individual IDs obtained at different target points.

Our approaches to Brand's algorithm illustrate how to deal with these issues in a suitable way. The next section explains the settings used to choose the target point.

4.2.1 The choices of the target points

Our initial aim is to identify some suitable target points for Brand's algorithm. The key question is over which target points this averaging is performed. The main issue that one should be aware of is that points close to the boundaries will lead to smaller estimated IDs. In order to avoid sampling from boundary points, one needs to identify a set of reasonably central target points. We propose two settings as follows:

- Setting **A**: This setting considers only potential target points x residing in the region
$$\{x|\ \hat{g}(x;\mathbf{H}) > c\},$$
where c is a density 'threshold' above which data points are considered to be central (with $\hat{g}$ being a kernel density estimator applied onto the data Z, see Section 1.2 and Subsection 1.3.4). While several choices of c are justifiable, we used 75% of the maximum density, i.e. $c = 0.75 \times \max\{\hat{g}(x_i;\mathbf{H})|\ 1 \leq i \leq N\}$, which achieved a good compromise between capturing sufficient structure and dismissing boundaries. A convenient sample, with respect to the number of data observations, of size 10 or 20, unless stated otherwise, can then be chosen from this region, and the median of the obtained values gives the overall ID estimator. We will illustrate in Section 4.6 that the number of sampled target points does not affect strongly the estimate of dimensionality.

- Setting **B** (just for testing the method): The principal curves are smooth curves through 'the middle of the data cloud' so they should do a good job in identifying central points. Then, if one has a prior evidence (e.g: from a visual impression) that the ID of a data set is approximately equal to one, then one may find central points through the 'local centers of mass' of a local principal curve (LPC) using function **lpc** in the 'LPCM' package [25]. In the case of 'LPC', the smoothing parameter is the bandwidth h that controls the degree of smoothing. This technique is not applicable for all data structures, because in some cases the principal curve does not fit well.

Comparing two settings, setting A works well for all data structures, as will be shown in the next chapter. In addition, this setting alleviates issue 1. In the following section we propose two variants of Brands algorithms which try to estimate the loglog curve, and, then we extract the ID locally under this scenario.

4.2.2 Variants of Brand's algorithm

Theiler [85] stated that ID estimation always requires some sort of averaging. While for global methods the averaging happens implicitly, for local methods this has to be done retrospectively using the 'local' IDs estimated at several target points. We use this technique with our new approaches to improve the practical implementation of Brand's algorithms.

Dip method

In order to obtain the intrinsic dimension, Brand proposed using the derivative function $G(r)$ which implies that the first peak in the function $G(r)$ is inspected. Practically, we found that the intrinsic dimension can be obtained by the inverse function of $G(r)$ which means direct use of the derivative

$$H(r) = \frac{\partial\left(\log N(r)\right)}{\partial\left(\log r\right)}, \tag{4.2}$$

which is easier to interpret, implement and alleviates issue 4.

Then it becomes obvious that finding the first *peak* of $G(.)$ is equivalent to identifying the first *dip*, say r_0, of $H(.)$. Note again that at the local linear scale, i.e. in a neighborhood of r_0, one has

$$N(r) \propto r^d,$$

or we can write

$$N(r) = \ c \cdot r^d.$$

where d is the intrinsic dimension and c is constant. Applying the logarithm to the this equality, we get

$$\log N(r) = \ \log(c) + \ d \log(r). \tag{4.3}$$

By substituting into the derivative operator $H(.)$, one finds that, at $r = r_0$,

$$\begin{aligned} H(r_0) &= \left.\frac{\partial \ (\log N(r))}{\partial \ (\log(r))}\right|_{r=r_0} \\ &= \left.\frac{\partial \ (\log(c) + d \ \log(r))}{\partial \ (\log(r))}\right|_{r=r_0} \\ &= \left.\frac{\partial \, (\log(c))}{\partial \, (\log(r))} + d \, \frac{\partial \, (\log(r))}{\partial \, (\log(r))}\right|_{r=r_0} \end{aligned}$$

so that

$$H(r_0) \ = \ d.$$

Therefore, if the process $H(r)$ takes a dip at r_0 then the ID is given by the value $H(r_0)$. In practice, the derivative $H(r)$ can be estimated by applying a local polynomial smoother of degree 2 onto the function of $\log N(r)$ versus $\log r$. We used `R` function **locpoly** in the 'KernSmooth' Package [74]. The local polynomial fitting is a nonparametric method with a kernel weight. It can be used to estimate either density, regression function or their derivatives. The degree of smoothing is determined by the bandwidth of the local polynomial, and it is chosen such that the curve passes well through the central part of the curve. Moreover, if the smoothing parameter 'bandwidth' is very small it produces a wiggly curve, and if the bandwidth is too big the resulting curve is very smooth. Therefore we choose a bandwidth of derivative higher than the local polynomial estimate to produce a smooth curve. We suggest a bandwidth parameter 0.15 unless stated otherwise. This bandwidth should work well universally provided that the data is scaled.

Next the ID is obtained by tracing the first derivative of the local polynomial curve and looking at the first dip in it. The intrinsic dimension is determined by the value of this dip on the vertical axis, which alleviates the issue 3. It is noted that the first derivative function might be thought of as the slope of function of the original graph. It also studies the relative change $\frac{\partial\,(\log N(r))}{\partial\,(\log(r))}$ of $N(r)$ when increasing or decreasing r by small value $\partial \log(r)$.

Regression method

This method uses linear regression to fit a line onto the curve of $\log N(r)$ versus $\log r$. To motivate this method, start again from (4.3), but consider now, similarly as for the fractal method, the limit for $r \longrightarrow 0$ instead of the derivative at $r = r_0$. Then

$$H(r) = \lim_{r \longrightarrow 0} \frac{\log N(r)}{\log(r)} \;=\; \lim_{r \longrightarrow 0} \frac{\log(c) \;+\; d\, \log(r)}{\log(r)}.$$

$$H(r) = \lim_{r \longrightarrow 0} \frac{\log(c)}{\log(r)} + d\,\frac{\log(r)}{\log(r)} = d. \tag{4.4}$$

So, taking the limit $r \longrightarrow 0$ also extracts the ID. This shows, in comparison with (4.6), that the same methods that are used to extract the ID from the correlation integral can in principle be used here as well, but using $N(r)$ in lieu of $C(r)$. Formalizing the loglog method [8] known from fractal ID estimation, the ID is estimated as the slope b of

$$\log(N(r)) = \; b\; \log r + \; a,$$

using a reasonable default range of small values of r. The conceptual downside of this method is that the neighborhood of r_0 in which (4.3) is valid almost certainly does not extend until $r = 0$, so derivation (4.4) is only of approximation character. Furthermore, this method comes with all problems associated with the estimation at $r = 0$ mentioned earlier in the context of fractal methods.

It should also be emphasized that the two approaches, Dip method and Regression method, are *local methods*, which need to be repeated for each target point, and then averaged over all target points. We summarize the implementation of Brand's algorithm in the following section.

4.2.3 Summary: Computation of Brand's charting manifold

- Step 1: Begin with the target points that are selected by one of our settings (see Section 4.2.1).

- Step 2: Choose a suitable range of radius r, where the radius expands for every target point. Practically our software provides a function that computes the minimum value of the radius that contains at least two points, to avoid an empty ball, while the maximum value of r is holding all data points. In addition, if the user chose setting **B** to select the target point, we provide a function that computes the distances matrix for all LPC points to keep away from boundary point.

- Step 3: For each value of data points, calculate an Euclidean distance between the data points and a selected (target) point, i.e.

$$\|x_i - x\|\,,\ \text{where}\ i = 1, \cdots, N. \tag{4.5}$$

  Then count the number of data points inside the ball to get $N(r)$, and sketch the loglog plot.

- Step 4: Estimate the intrinsic dimension 'locally' by using one of our approaches of variants of Brand's algorithm, see Subsection 4.2.2.

- Step 5: Repeat steps 2-4 at different target points in order to look at the intrinsic dimensionality development along the data cloud.

Finally, in order to capture the intrinsic structure of data, the median overall ID estimates is computed while reducing boundary effects. If desired, one can round the value to the nearest integer. To sum up the issues discussed in Section 4.2, the step1 and 2 attempt to overcome issues 1 and 2. Step 3 alleviates the issue 3. The Dip regression technique alleviates the issue 4. By taking the median over all ID estimates one deals with issue 5.

4.3 Intrinsic dimension via correlation dimension

Correlation dimension is used to obtain the fractal methods that describe the attractor dimension. It is a global dimensionality estimation method. This method differs from Brand's charting manifold by counting the pair distances rather than points. Again, the correlation dimension is defined as

$$d_{cor} = \lim_{r \to 0} \frac{\log(C(r))}{\log(r)}, \tag{4.6}$$

where according to the GP method, the quantity $C(r)$ is obtained as:

$$C(r) = \lim_{N \to \infty} \frac{2}{N(N-1)} \sum_{i=1}^{N} \sum_{j=i+1}^{N} I\left(\|x_j - x_i\| \leq r\right) \tag{4.7}$$

The implementation of the correlation dimension method requires consideration of the following factors:

1. Original data dimensionality.
2. The correlation between the variables.
3. The determination of range of r.
4. The portion of distance pairs that are used for calculation (the sample size needs to be so large).

In addition to these factors the core problem is that the practical computation of the correlation dimension is far from straightforward. This is due to the fact that the correlation integral needs to be estimated for a ball tending to 0, and there are no data positioned within that ball. Then, one needs to decide on a suitable range of values of r which is used to arrive at an estimate of the ID [85]. Furthermore, it is obvious that in practice infinite sample sizes cannot be achieved when the limit of $N \to \infty$ in Eq.(4.7) is concerned.

Our approaches try to minimize the demand on those factors and provide the best result. We try to capture the distance pairs of $C(r)$ in a more effective way which is consistent with the GP method. The algorithms achieve the estimation of the ID of a given data set at radius $r = 0$. The developed algorithms are Intercept method, Slope method and Polynomial method. While the Slope method is effectively an implementation of the loglog technique described above, which makes use of the approximately linear part of the correlation integral curve, the other two methods are entirely new and tackle the problem by direct exploitation of the features of the function $\frac{\log(C(r))}{\log(r)}$ and $C(r)$, respectively. All three approaches are based on the concept of linear regression. The improved methods are described in the following subsection.

4.3.1 The implementation of Correlation dimension

Intercept method

It is obvious that the radius r can not be equal to zero, which would mean that there are no data points in the circle, yielding 'NAN' for $C(r)$. The Intercept method estimates the fractal dimension not through direct evaluation of $C(r)$ at $r \approx 0$, but through linear extrapolation of the graph $(r, D(r))$, where $D(r) = \frac{\log(C(r))}{\log(r)}$. In practice, the curve $D(r)$ is plotted versus the radius r. Then a grid of values of r, say $r_j, j = 1, \ldots, s$ is chosen which is positioned close to 0 and contains a sufficient number of data points. In practice choices like $0.3 \leq r \leq 0.5$, with a grid size of $s = 30$, work well. Hence, it is only necessary to compute the correlation integral for a portion of data pairs which reduces the computational time.

This approach is motivated through similar ideas proposed by Rummel [77], who suggested backwards extrapolation to obtain regression estimates under covariate measurement error ('SIMEX'). Following this idea, we predict the intrinsic dimension by extrapolating a linear regression line (fitted to the values $(r_j, D(r_j)), j = 1, \ldots, s$) to $r \to 0$. The intrinsic dimension is then obtained as the intercept of the fitted linear equation. Specifically, consider a linear regression with least squares estimator a (intercept) and c (slope). Then the correlation dimension can be approximated as

$$D(r) = a + c\, r,$$

which at $r = 0$ gives

$$d_{cor} = D(0) = a.$$

Using this method, the fractal dimension is defined as the intercept part of a linear equation at $r = 0$. It is obvious that for a loglog plot we can not calculate the correlation dimension at $r = 0$.

Through experimental analysis it is shown that this approach improves the correlation dimension calculation for any type of data set. In addition, this approach requires fewer data points and less demand on sample size.

Slope method

In this section we exploit the previously stated properties of the loglog curve of the correlation integral. Hence, suppose the high-dimensional data set Z has an intrinsic dimension d. If the sample size is large enough then the number of distance pairs will increase due to the increase of r, and since $C(r)$ is a function of r, then as r increases $C(r)$ will increase proportionally with r^d. As we illustrated in Subsection 3.5.1, At $r \to 0$, $d_{cor} = d$, which means that the correlation dimension is a good estimate of the intrinsic dimension of the corresponding data set.

Now, to obtain the estimate of intrinsic dimension, we plot the curve of $\log(C(r))$ versus $\log(r)$ and the slope value is computed using a simple linear regression method which fits a line on the curve of $\log\ (C(r))$. This is done by assuming that the equation of the regression line is:

$$\log(C(r)) = b\ \log(r) + a,$$

where a is the intercept and the slope of the equation (b) is the estimate of the intrinsic dimension, i.e. $d_{cor} = b$. For the choice of interval in which the linear regression is fitted, we recommend $0.3 \leq r \leq 0.5$ again.

Polynomial method

This section provides a potential model for the correlation integral based on the relationship between the correlation integral $C(r)$ and the radius r. We develop an approach in which $C(r)$ is explicitly modelled through a higher–order polynomial, considering the following condition:

- at $r = 0, \Rightarrow C(0) = 0$.

We state the following general result (see appendix for proof): For a polynomial with degree p, let $C(r) = a_p r^p + \cdots + a_2 r^2 + a_1 r + a_0$, and subject to constraint $C(0) = 0$: one has

1. If a_1 exists then $d = 1$,

2. For $a_1 = 0$, then $d = 2$,

3. For $a_2 = a_1 = 0$, then $d = 3$,

4. For $a_{p-1} = \cdots = a_2 = a_1 = 0$, then $d = p$.

The correlation dimension can be obtained using multiple linear regression (e.g. function `lm` in `R`), and as a default we assume that $C(r) = a_4 r^4 + a_3 r^3 + a_2 r^2 + a_1 r$ (the polynomial degree would need to be increased in order to detect IDs with $d \geq 5$). Then one examines the significance of parameters by t-test and the ID is the first significant parameter. In practice, we recommend leaving the significance level of this test unspecified and determining the ID by the *most* significant parameter, that is, the parameter with the largest t-value or the smallest p-value.

4.3.2 Summary: Computation of Correlation dimension

The following shows how the ID is computed via correlation dimension approaches.

- Step 1: Define the range of radius r as follows. For the Intercept method and the Slope method we choose a range of r between 0.3 and 0.5 which is a portion of the data range. At that range the curve of $C(r)$ often looks roughly linear and we can avoid outlying values. For the Poly method, we define a function which scans all the distances between two data points to determine the minimum radius r such that the circle holds at least two points. This step is important before picking a sequence of r to avoid the interruption of process, otherwise one gets 'NAN' at $C(r)$, when calculating the correlation dimension.

- Step 2: For a specific distance r, count the number of pairs of data points, such that the Euclidean distance between two data points is less than r, i.e.

$$\sum_{i=1}^{N} \sum_{j=i+1}^{N} I\left(\|x_j - x_i\| \leq r\right).$$

- Step 3: Calculate the correlation integral $C(r)$ as a function of r, for fixed N

$$C(r) = \frac{2}{N(N-1)} \sum_{i=1}^{N} \sum_{j=i+1}^{N} I\left(\|x_j - x_i\| \leq r\right).$$

- Step 4: Draw various plots which have been generated according to the methods applied, and then the ID for the data set is obtained.

The step 3 to 4 are applied on Intercept method and Slope method, while for Polynomial method we apply the steps in this way.

- Step 3: Calculate the correlation integral $C(r)$ as a higher–order polynomial

$$C(r) = a_4 r^4 + a_3 r^3 + a_2 r^2 + a_1 r$$

  This is done by using `R` function **lm** with order 4 as a default [74].

- Step 4: Determine the ID by looking at the *most* significant parameter.

4.4 Intrinsic dimension via a local version of a global method

This section introduces an alternative approach that estimates the ID if the global ID methods are implemented on the subregion of the data set. This method attempts to overcome several issues such as bias, computational cost and dependence on data structure. It is noted that the correlation dimension provides the smallest bias [60]. The objective of the localized global approach is to improve the algorithm based on a local ID method (such as Brand's algorithm), which could significantly reduce the negative bias. This can be justified as follows.

Let us tentatively define $\tilde{C}(r)$ as the number of *pairs* situated within a ball of radius r around a certain target point x. Then

$$\tilde{C}(r) = \left(\begin{array}{c} N(r) \\ 2 \end{array} \right) \propto \frac{r^d(r^d-1)}{2} = O(r^{2d})$$

would (at signal scale) increase with r^{2d}, so that the resulting intrinsic dimensionality estimate obtained through this route *would need to be divided by 2.* We do not pursue this route further in this manuscript, but this aspect is important for our understanding the difference charting makes to the correlation dimension. We used this concept to illustrate new approach (charting with pairs) which will be introduced next.

By dividing the data region into several separated subregions, a correlation dimension approach can be derived from the data of each of these disconnected subregions. With respect to the number of disconnected subregions, this would produce as many ID estimates as the number of subregions. The process is completed if the number of remaining data points is zero or less than 3. The detailed explanation of the localized global approach in this section is structured as follows. Subsection 4.4.1 explains the strategies of estimate the ID via a local version of a global method.

4.4.1 Computation of localized correlation integral method

In this section we introduce two possible ways of implementing the localized global method. We illustrate two possible techniques: Charting by pairs and Localized correlation integral methods. In practice we have actually only implemented the second one.

This section explains how to construct the subregions and how many disconnected regions could be considered. In practice, dividing the operating data range into disconnected regions can be conducted by directly analyzing the data set. This analysis is based on determining the number of data points that lie within a specific radius for each subregion. The manner of ID estimation via localized global approach is explained as follows.

Charting with pairs method

- Step 1: Choose a starting point as in Subsection 4.2.3.
- Step 2: Choose a suitable range of radius r, where the radius expands for every starting point. The range of r is selected by determining the minimum radius that contains at least two points and the maximum value of r contains all data points.
- Step 3: For each value of data points, calculate an Euclidean distance between the data points and a selected target point as Brand's algorithm implementation as (4.5), i.e.
$$\|x_i - x\|, \text{where } i = 1, \cdots, N. \tag{4.8}$$
- Step 4: Count the number of data pairs inside the ball, and compute
$$C(r) = \lim_{N\to\infty} \frac{2}{N(N-1)} \sum_{i=1}^{N} \sum_{j=i+1}^{N} I\left(\|x_j - x_i\| \le r\right). \tag{4.9}$$
- Step 5: Obtain ID by using one of our approaches of variants of Brand's algorithm, see Section 4.2.2.
- Step 6: Iterate steps 2-5 at different target points. Then compute the median of ID estimates.

The dimensionality estimation via Charting with pairs method should satisfy the concept of Subsection 3.5. As it is not entirely clear how to connect (4.8) and (4.9), we provide an alternative approach to implementing this concept.

Localized correlation integral

The idea of this approach is prompted through similar ideas suggested by Fukunaga [32], who proposed an algorithm that obtained ID locally by minimizing the local region size until reaching the limited dimensionality.

- Step 1: Choose some arbitrary points.
- Step 2: Choose a suitable range of radius r that contains sufficient data points in the neighborhood of one of the arbitrary points (x), such as 20 - 30 points.
- Step 3: Carry out the correlation dimension methods, Subsection 4.3.1, in this neighborhood.
- Step 4: Construct a matrix of a temporary data set which consists of all the original data points. Discard the points that are in the neighborhood (step 2).
- Step 5: Repeat steps 1-4 by using a temporary data set.

The process is iterated until the temporary data set is empty, or it only contains a few disconnected points. The considerations provided at the beginning of this subsection would suggest to divide the ID result by 2. Further discussion is provided in Chapter 5. The ID is obtained by computing the median over all ID estimates which is consistent with other approaches. It is important to note that the value of ID varies depending on the counting of data points in each subregion and the selection of the arbitrary points. It should be emphasized that it is important to select sufficient data points which prevent the crash of the process, i.e. if the remaining subregion contains too few data points, the process will stop.

4.5 Computation of maximum likelihood estimator

Suppose k is the number of nearest neighbors and $T_k(x)$ is the Euclidean distance from the data point x to its $k-$th nearest neighbor in the sample. We apply the maximum likelihood estimation (MLE) algorithm (see Subsection 3.3.5) as outlined in the following steps:

- Step 1: Determine a suitable range of k. We choose k which is small enough to have enough points in the sphere, and k is increased sequentially.
- Step 2: Define a function which computes the distance from x to each different data point and define the distance matrix.
- Step 3: Define a function that calculates MLE for dimension d as,

$$d_k(x) = \left[\frac{1}{k-2} \sum_{j=1}^{k-1} \log \frac{T_k(x)}{T_j(x)} \right]^{-1}.$$

- Step 4: Obtain the ID locally at every data point by computing the average dimension estimation within data sphere as,

$$d_k = \frac{1}{N} \sum_{i=1}^{N} d_k(x_i),$$

- Step 5: The process, step 2-4, is repeated for a set of values of k, say $k_1, \cdots, k_z$ within the data range. Practically the suitable range of k is 10 to 20
- Step 6: We obtain the ID over the entire data set by computing the median of d_k over a range of different k to neglect the effects of k.

In step 6 we propose to use the median, in contrast to Levina and Bickel's algorithm for MLE estimation who use the mean. The median is applied to derive an ID consistent with our approaches via the correlation dimension and Brand's algorithm.

Note that for the choice of range of k, it is important not to choose a very small k which could lead to unreasonable estimates and not to choose a very large k which could result in an estimate with a negative bias.

4.6 Experiment on Artificial data sets

In this section we discuss the effectiveness of our techniques, variants of Brand's algorithm and correlation dimension methods, and the computational results for artificial data sets. These are Spiral data and Swissroll with known intrinsic dimensions which equal 1 and 2, respectively. In Subsection 3.3.6 and 3.5.2 we implemented MLE (local ID method), and also global ID methods (linear PCA, Kernel PCA and MDS) to those data sets.

For the variants of Brand's method, known as the Dip method and the Regression method, the sequence of the radius r is selected such that the lower point is the minimum r that contains at least two data points, while the upper point holds all the data points.

For the implementation of correlation dimension, Intercept method and Slope method, the reasonable sequence of r is 0.3 to 0.5. In contrast, for the Polynomial method, the lower point is the minimum value of r that contains at least one distance pair, which is consistent with the minimum r selected for Brand's algorithm, and the upper point of r equals 1.

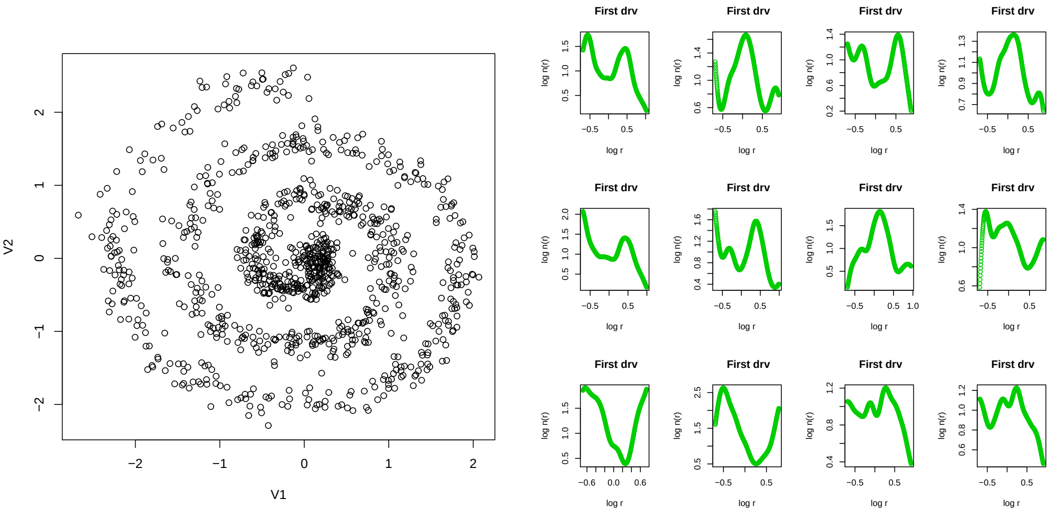
Finally, a comparison is made with the principal component analysis (PCA), Kernel PCA, MDS and the MLE methods. Next, the analysis begins with the implementation of methods on Spiral data.

4.6.1 Spiral data

As we mentioned in Subsection 3.3.6, the Spiral data consists of two variables with 300 data points. Figure 4.1a illustrates that the intrinsic dimensionality of data is equal to 1, while the ID estimate via MLE method (Median ID=1.84) at $k = 10, \ldots, 20$, and the implementation of linear PCA and Kernel PCA in Subsection 3.5.2 indicates that the ID equals 2.

Now, we compare these results to the estimated dimensionality via Brand's algorithm and correlation dimension estimation methods.

For Brand's algorithm implementation, the intrinsic dimension is derived using the *Dip method* and the *Regression method.* (a). For the application of the Dip



(a) (b)

Figure 4.1: (a) A 2D scatter plot of scaled Spiral data, (b) The ID estimate of Spiral data via Dip method at 24 target points. .

method, we consider the target points according to the highest–density–criterion outlined earlier. The first derivative estimator is found using a local polynomial smoother with the bandwidth 0.15 for a sample of size 24 chosen from the higher density points, as shown in Figure 4.1b. The median of all different ID estimations is 0.9138581. To demonstrate the effect of the sample size of target points, we select 40 target points from higher density points. The median of all different ID estimations is 0.8257418, which clarifies that the ID estimates are not influenced strongly by the sample size of target points.

(b). For the Regression method, we estimate the ID for each hyper–sphere of previous (24) target points by fitting a linear regression. The local ID is obtained from the slope of the regression line. Then, the ID is estimated by computing the median of the ID estimates, which is equal to 1.019648. Both techniques provide results which are lower than the MLE result.

Next, the ID is obtained via the correlation dimension:

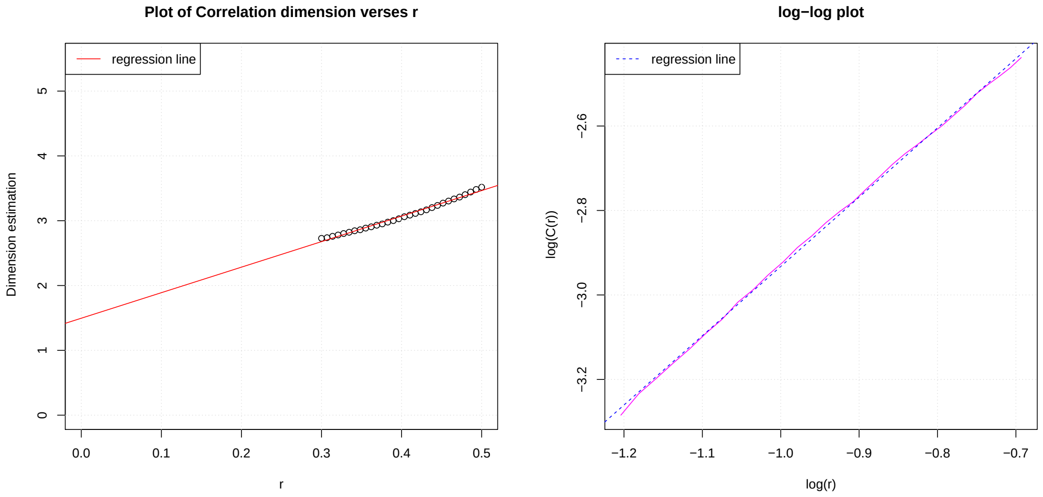
(a). *Intercept method* - We plot $c(r)$ versus versus r. Figure 4.2a shows that the curve of the correlation dimension is mostly linear in the chosen range of r. Figure

4.2a displays the fitted regression line $D(r) = a + c\,r$ on the correlation dimension curve. Therefore ID = 1.50, which is the intercept value in the linear equation of $y = 1.495580 + 3.939566\ (r)$.

(b). *Slope Method* - The linear regression is fitted through the curve of $\log(C(r))$ in the loglog plot as shown in figure 4.2b. The linear equation is $y = -1.292517 + 1.640014\log(r)$, so the intrinsic dimension is equal to $b = 1.64$. The result is reasonably close to the Intercept method.

(c). *Polynomial method* - The ID is derived by considering the largest t-value of parameters. For a polynomial of degree 2, one observes from Table 4.1 that the t-value for a_1 is slightly larger than a_2, so the intrinsic dimension of 1 is clearly identified.

We find that the techniques (Dip, Regression and Polynomial methods) arrive at sensible results which broadly agree with each other, and are consistent in line with the visual impression. While the result using Intercept and Slope method are consistently with MLE, linear PCA and Kernel PCA methods. All these methods provide an overestimated ID.



(a) (b)

Figure 4.2: Spiral data; (a) the plot of $D(r)$ versus r which is roughly linear for a reasonable range of r, (b) the log-log plot of correlation integral versus r.

```
-------------------------------------------------------------------------
Coefficients:
        Estimate Std. Error t value Pr(>|t|)
re      0.105756   0.006520   16.22 9.08e-16 ***
I(re^2) 0.122408   0.008282   14.78 9.43e-15 ***
---
Signif.codes:0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 0.005282 on 28 degrees of freedom
Multiple R-squared: 0.9982,     Adjusted R-squared: 0.998
F-statistic:  7577 on 2 and 28 DF,  p-value: < 2.2e-16
-------------------------------------------------------------------------
```

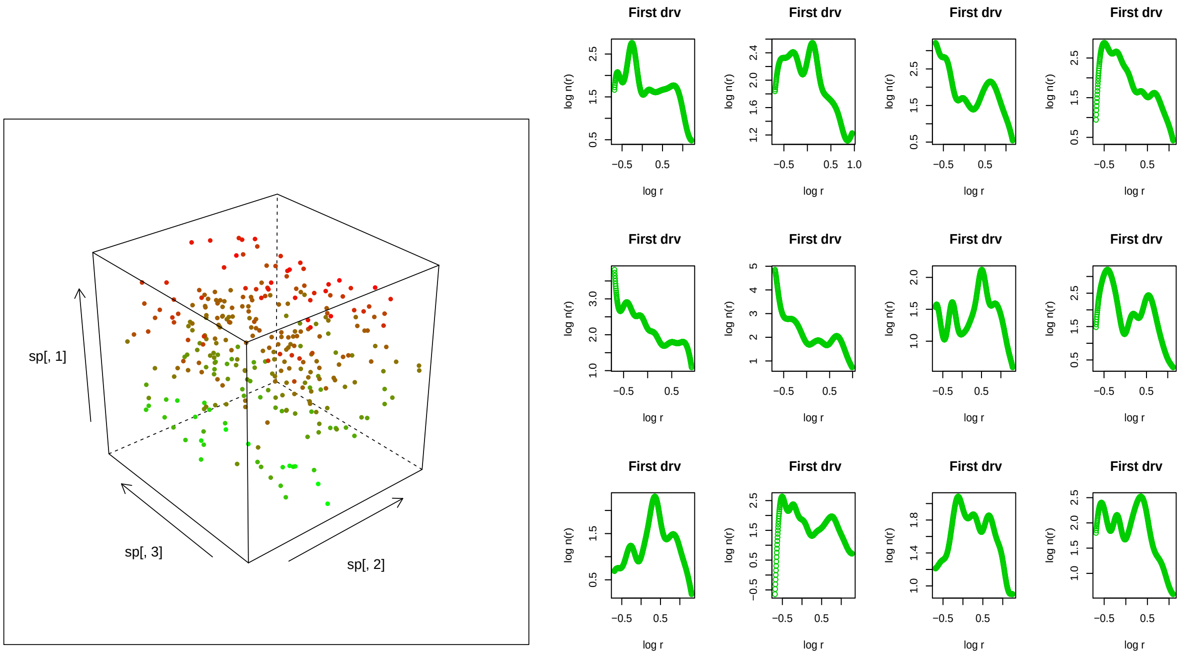
Table 4.1: Spiral data: the result of fitting a polynomial of degree 2.

4.6.2 Swissroll data

The Swissroll data consists of three variables with 300 data points, see Subsection 3.3.6, with known intrinsic dimensionality equal to 2. Both of the MLE method at $k =$ 10, . . . , 20 (Median ID=2.51), and linear PCA in Subsection 3.5.2 provide estimated IDs equal to 3. Using the MDS method, the ID is equal to 2.

We now compare these results to the estimated dimensionality via Brand's algorithm and correlation dimension estimation methods.

The Intrinsic dimension estimation obtained using Brand's method; Firstly, for the implementing of the *Dip method*, we choose a sample of size 24 of target points according to the highest–density–criterion outlined earlier. The ID is estimated for each target point by computing the first derivative with bandwidth 0.15 as shown in Figure 4.3b. The median of all different ID estimations is 1.544308. Secondly, with the *Regression method*, the ID is estimated by fitting the linear regression method on the previous target points and determining the slope of the regression. Then the ID is derived as the median of the ID estimates which is equal to 1.751609. This



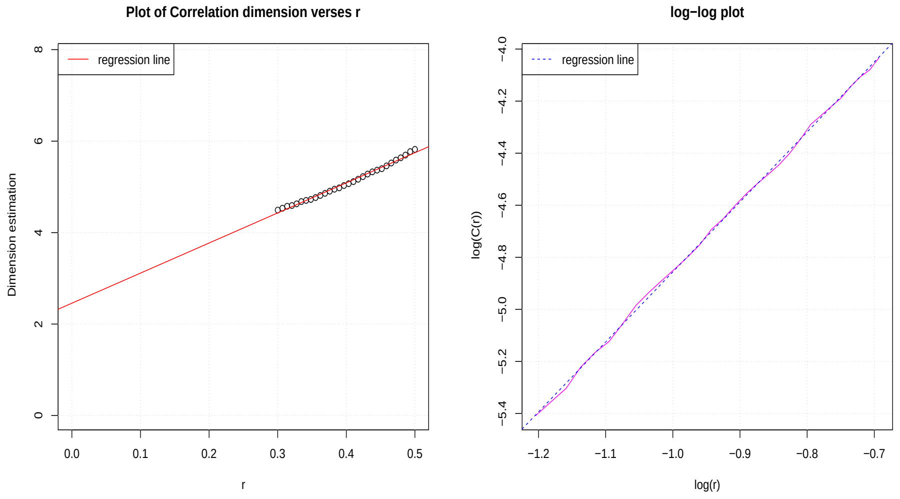
(a) (b)

Figure 4.3: (a) A 3D scatter plot of scaled Swissroll data, (b) The ID estimate of Swissroll data via dip method at 24 target points.

value is close to the dimension value estimated by the Dip method. Both methods provide reasonable ID compared to MLE, which provides overestimated ID.

Next, the intrinsic dimensionality is estimated via correlation dimension: Firstly, the *Intercept method* implementation. We study the correlation dimension curve with radius r. Here, as shown in Figure 4.4a the curve of correlation dimension looks reasonably linear in the chosen range of r. Figure 4.4a displays the fitted regression line $D(r) = a + c\,r$ on the correlation dimension curve. Then, ID $= 2.46$ which is the intercept value in the linear equation of $y = 2.456139 + 6.581708\ (r)$. Secondly, we test the *Slope method*. The plot in Figure 4.4b displays the curve of $\log(C(r))$ versus $\log(r)$ with a fitted linear regression. Therefore, the estimated intrinsic dimension is equal to 2.69 where the linear equation is $y = -2.167556 + 2.688537\ \log(r)$. This value is slightly larger than the dimension value estimated by the Intercept method.

Finally, using the *Polynomial method*, the ID is derived via a series of $t-$ tests



(a) (b)

Figure 4.4: Swissroll data; (a) The correlation dimension curve with range of r, (b) The log-log plot of correlation integral versus r.

on the model parameters. We assume that the correlation integral is modelled by a polynomial of degree 3. The results are shown in Table 4.2 with the upper value of r equal to 1. From table 4.2, the most significant parameter is a_2, and hence, ID $= 2$.

We find that there is some discrepancy in the observed dimension estimates. While the intuitive scree-plot based solution of ID $= 2$ is backed up by the Polynomial method, we observe a larger value of $\approx$2.4 and 2.6 via the Intercept and Slope method, respectively. In addition, we obtain smaller values of 1.54 and 1.75 from the Dip method and Regression method, respectively.

4.6.3 Discussion of bias

In this section, we discuss the bias of the estimators regarding to our approaches. The bias is defined as the difference between the estimator's expected value and the true value of the estimator. In most cases it would be desirable to use the estimator with less bias. It is important to note that all ID estimation methods suffer from bias.

In practice, we take 100 samples of 200 data points generated from Swissroll

```
-------------------------------------------------------------------------
Coefficients:

         Estimate Std. Error t value Pr(>|t|)

re      -0.016099   0.001331 -12.097 2.06e-12 ***

I(re^2)  0.106844   0.004060  26.318  < 2e-16 ***

I(re^3) -0.007204   0.002955  -2.438   0.0217 *

----------

Signif.codes:0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 0.0004309 on 27 degrees of freedom

Multiple R-squared: 0.9999,     Adjusted R-squared: 0.9999

F-statistic: 7.688e+04 on 3 and 27 DF,  p-value: < 2.2e-16

-------------------------------------------------------------------------
```

Table 4.2: Swissroll data; the result of fitting a polynomial of degree 3.

data. Then we compute the mean value of the ID estimates which are derived by using Dip, Regression, Intercept, Slope and Polynomial method. In addition, the ID is obtained via MLE by computing the median of d_k over a range of different k, as shown in Section 4.5. We provide a box-plot of the ID estimates for 100 Swissroll data sets. The plot in Figure 4.5 illustrates that our methods obtain a reasonable estimate of the intrinsic dimension, with the Intercept method achieving results (median: 2.10301) which are closer to 2 than the Slope method (median: 2.386272), while the median of the IDs via MLE is 2.798521. Comparing those methods with the other local approaches via the Dip method and the Regression method, the median ID using the Regression method is 1.696434, which is closer to 2, while the median using the Dip method is 1.427049, which means it provides underestimated ID. In contrast, the Polynomial method has returned ID = 2.

Figure 4.5 shows that the Intercept method has the smallest bias compared to all other our approaches, even though it has slightly a larger variance (variance(IDs)= 0.05) than other approaches. The local methods, Dip and Regression methods, provide a negative bias. We observe that the bias in the Dip and Regression methods (Local method) due to the limited data size. One could consider the under–

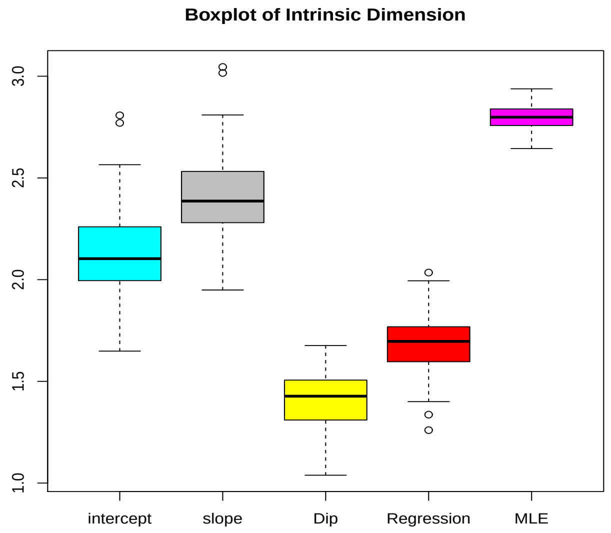


Figure 4.5: A box plot of ID estimates via Intercept, Slope, Dip, Regression and MLE methods of 100 data sets generated from Swissroll data.

estimation is simply a feature of the local methods, which estimate the topological dimension along the data.

On the other hand, the Intercept, Slope and Polynomial methods have a smaller computation time than the Dip and Regression methods. It is clear that for the Dip and Regression methods the technique should obtain the ID for each sphere, then average overall ID estimates. To sum up, it is noted that the ID local method always provides a lower bound of ID estimates. Furthermore, the ID global methods provide an estimate greater than the estimate provided by local methods. We can conclude that the Intercept method provides a suitable result with a small bias and less computation.

4.7 Software

The statistical computing software R [74] has been used to execute all the practical implementations of the dimensionality estimation. The R programs are used to compute the examples and the simulation study. The code of the principal manifold is taken from package ‘lpmforge’ (unpublished) version $0.0-8$. The examples data are taking from the following Packages:

- Horse mussels data from package forward [70].
- Oceanographic data from package LPCM [25].
- Gaia data from package LPCM [25].
- Fuel consumption data and industrial melter data are provided in txt file.
- Spiral data is provided in dta file.

The industrial melter data was provided by Dr. Uwe Kruger. The spiral data was provided by Dr. Balász Kégl (https://www.lri.fr/ kegl/researchUdeM/research/pcurves/). The necessary modification to the programs is made as in demand. Our own code explaining how we implemented our new approach is available in http://www.maths.dur.ac.uk/∼dm

4.8 Conclusion

In this chapter the practical implementation of dimensionality estimation was explored. The intrinsic dimension was estimated via Brand’s algorithm by scrutinising the growth point process, which counts the number of points in hyper-spheres. Using correlation dimension the intrinsic dimension was obtained via a pairwise distances algorithm, which counts the number of point pairs that are closer to each other than a given radius.

The ID was obtained via the MLE method by investigating the number of observations falling in a small sphere. Thus we can deduce that the MLE method has properties of both local and global methods when used to estimate ID. This is to some extent true for Brand’s algorithm as well, though this requires the selection of a few target points which make such methods less 'global' in comparison to the MLE method, for which each data point is a target point.

The correlation method and the MLE method require large sample sizes for high-dimensional data. Maximum Likelihood estimation is influenced by the number of nearest neighbors. Novel approaches for the implementation of these techniques were supplied.

The intrinsic dimension was estimated locally via Brand's algorithm. Two settings were provided to select the target point and suggest the range of the radius of the hyper-sphere. Two approaches to estimate ID from the loglog curve were proposed: the Dip method and the Regression method. The intrinsic dimension of the data set was determined by computing the median over all IDs estimates.

Regarding our approaches for computation of the correlation dimension, we put forward three approaches: the Intercept method, the Slope method and the Polynomial method. In contrast to the Intercept and Slope methods, the Polynomial method provides an integer ID estimator (so, the estimated ID is not really 'fractal' in a strict sense). For the regression step, we suggested using an interval of r values ranging from the value of r that contains one distance pair as a minimum point and increases to the value point of $r = 1$. This ensures that the radia are close to 0 but hold sufficient data points.

Compared to our other approaches, the Regression, the Intercept and the Slope methods, the Polynomial method needs additional data points, because it fits a more complex model. Therefore, a larger upper r is needed in comparison to these other methods. We should note that increasing the polynomial degree beyond $p = 4$ sometimes leads to unclear results, since the higher–degree polynomials correlate in a complex manner with each other, which dilutes the distinctiveness with which the intrinsic dimension is identified.

We have observed the values of PCA-based ID to be often larger than those obtained by nonparametric ID estimation methods. Additionally, we found the IDs obtained by the global methods (Intercept, Slope) often to be more accurate than those by local methods (Regression, Dip), with the Dip method quite persistently underestimating the ID. The ML method produced generally reasonable ID estimates, which were often (atypically for a local method) close to the result by PCA, and sometimes even larger which may be a sign for a tendency to overestimate the true ID.

To sum up, it is noted that the ID local method always provides a lower bound of ID estimates. On the other hand, the ID global methods provide an estimate greater than the estimate provided by local methods. We can conclude that the Intercept method provides a suitable result with a small bias and less computation.

Chapter 5

Experimental Results

5.1 Introduction

In Chapter 4 we demonstrated our new approaches for computing dimensionality estimation via charting manifold and fractal-based methods, also the approach via a local version of a global method. In this chapter we discuss the computational results for data sets in multivariate space, and the effectiveness of our techniques.

To illustrate the performance of the methods under investigation, we provide simulation examples and applications to several experimental data sets. The experimental data sets describe different phenomena and are available in the literature, in R packages [74] [70] [25]. In addition we provide a recorded data set from an industrial glass melter process. When a subsample of the full data set (size N) is taken, then we denote the subsample size by n. Section 5.3 illustrates the application of the ID estimation approaches to Horse mussels data ($D = 4$), Oceangraphic data ($D = 3$), Airquality data ($D = 4$), Gaia data ($D = 19$) and Fuel data ($D = 4$). For Gaia data we take a sample of data points to simplify the implementation of the MLE method. A comparison of the experimental results with other methods is discussed in Section 5.4. Section 5.5 presents studies of simulation. An analysis of the industrial melter data ($D = 21$) is carried out in Section 5.6. Eventually the conclusion is presented in Section 5.7.

5.2 Preliminary concepts

We verify our methods on real data sets in multivariate spaces. All data are scaled to zero mean and unit standard deviation before implementation. We provide scatter plots of the data to represent the structure of the data sets. In practice, for the variants of Brand's method, known as Dip method and Regression method, the sequence of the radius r is selected such that the lower point is the minimum r that contains at least two data points, with the upper point that holding all the data points.

For the implementation of correlation dimension, Intercept method and Slope method, the reasonable sequence of r is 0.3 to 0.5. In contrast, for the Polynomial method, the lower point is the minimum value of r that contains at least one pair distance, which is consistent with the minimum r selected for Brand's algorithm, and the upper point of r equals 1.

As a proof of concept, for the Horse mussels data and Oceangraphic data, we implement the Dip method and the Regression method by using setting B to select the target points. Postulating that the data possesses an ID of about 1, one should be able to recover this one–dimensional structure using adequate dimension reduction tools. We use this setting only for validation of our approaches (Dip method and Regression method). When $d = 1$, the principal curve should pass through the middle of the data points and so be useful for identifying central points.

We also test the Localized correlation integral method on Horse mussels and Airquality data. The algorithm is iterated until the subregion set contains two data points. For simplicity, the data structure is partitioned into ten subregions with radius equal to 1. Each subregion contains n data points. The Intercept method (correlation dimension approach) is implemented on each subregion.

Eventually, a comparison is made with the principal component analysis (PCA) and the MLE methods.

For the PCA method we use `R` function **prcomp** [74] to produce a scree plot which provides the (linear, PCA-based) intrinsic dimension. The PCA is obtained via this function using a singular value decomposition of a scaled data matrix rather than the covariance matrix of a data set. Now, to plot this object we use one of the two functions **scree plot** and **plot** [74] which plot the variances versus the number of principal components. The option (scale=TRUE) indicates that the variables are scaled to have unit variance before the analysis. It should be emphasized that the intrinsic dimension (ID) arrived at through PCA method is usually larger (one could say an 'upper bound') than the nonparametric methods of intrinsic dimension estimation.

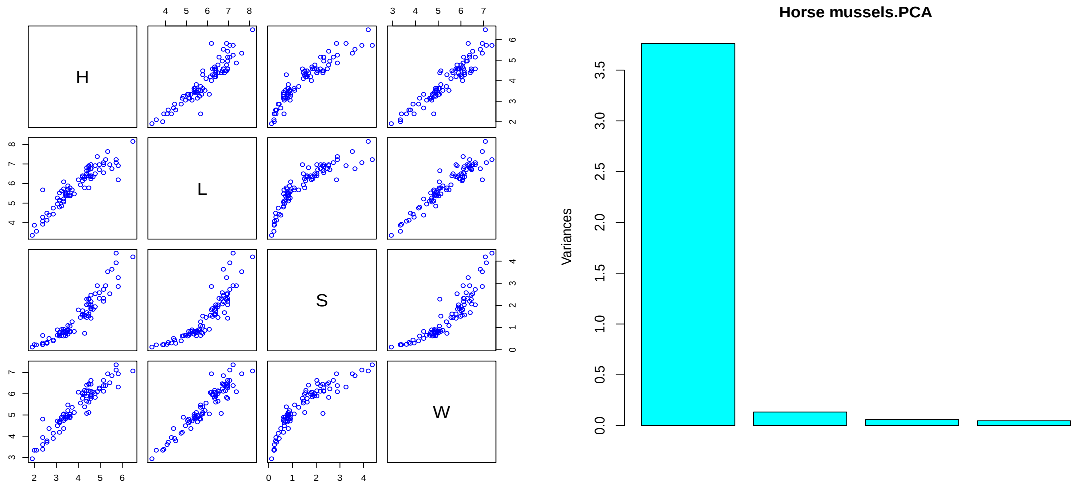
For MLE we use our practical implementation of the ID estimator by Levina and Bickel [60] (Section 4.5). We apply the algorithm of maximum likelihood estimation to different ranges of (k), where k is the selection of the number of nearest neighbors. In practice for small numbers of neighbors k the MLE algorithm provides an unreasonable value of dimension estimation. This leads one to infer that the algorithm has not worked yet. In addition the intrinsic dimension estimation is frequently low when k increases. We use a reasonable range of k between 10 and 20 as advised by Levina and Bickel [60].

The following section presents a description of data sets with the computational results for each set of data shown separately.

5.3 Applications

5.3.1 Horse mussels data

In this section we discuss the Horse mussels data (sampled from Marlborough Sounds, New Zealand) with 82 observations on five variables; shell width (W), height (H), length (L), mass (S), and the mass of mussels (M). The data is available in the package ‘ forward’ [70], and we will only consider four variables: height, length, mass and width. To gain an insight into the structure of the data, we plot the scatter matrix plot of the four (scaled) variables as shown in Figure 5.1a.



(a)

(b)

Figure 5.1: Horse mussels data; (a) Scatter plot matrix. (b) Scree plot of linear PCA.

We first estimate the dimensionality via linear PCA. Figure 5.1b illustrates the result of a principal component analysis on the (scaled) data set. The first and second components of PC explain 94% and 3%, respectively, of the total variance. Clearly, when performing linear dimension reduction via PCA, users decide the dimension according to how much variance they want to preserve. Hence, depending on this choice (common default choices would be 90% or 95%), we can conclude that the (linear) ID for this data set is 1 or 2, which matches the visual impression from

Figure 5.1a .

Next we estimate the ID using our approaches. Firstly, proceeding with the implementation via Brand's algorithm, we choose the target points according to the LPC setting (setting B in section 4.2.1). The LPC is fitted as shown in Figure 5.2a which is close to the scatter-plot of the raw data. The LPC is fitted with a starting point $x_0 = (0.970037, 1.343527, 0.4350951, 1.341437)^T$ and bandwidth $h = 0.2$. The fitted LPC is *one* curve through a $four-$dimensional space. As target points for the ID estimation we use the local means. Then,

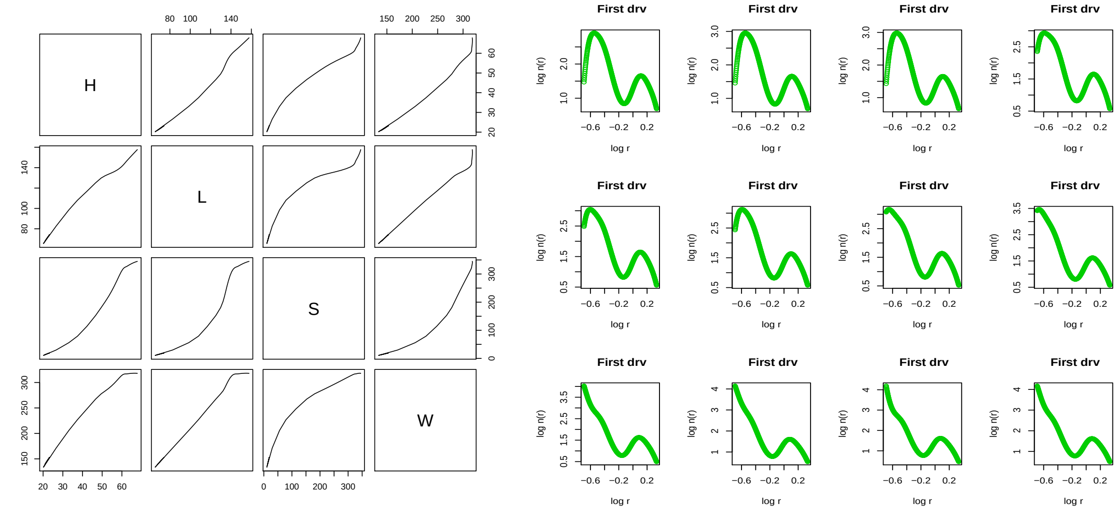
(a). Using the *Dip method*. The first derivative estimator is derived using a local polynomial smoother with bandwidth = 0.15. Each curve in Figure 5.2b represents the first derivative estimation for some selected LPC points. The median of all the different intrinsic dimension estimations is 0.8119179. **(b)**. Now, the implementation using *Regression method*. We estimate the ID for each hyper-sphere of previous target points. By fitting linear regression the local ID is obtained as the slope of the line. Then the dimensionality is derived by computing the median of the ID estimates which is equal 1.52486. Both methods provide a reasonable estimate of ID.

Secondly, estimate the dimensionality via correlation dimension.

(a). The implementation of the *Intercept method*. We start the implementation by studying the correlation dimension curve with radius r. Here Figure 5.3a illustrates that the curve is given by a grid on the right side, and the curve looks to be reasonably linear from 0.3 to 0.5. Figure 5.3a displays the fitted linear regression $D(r) = a + cr$ on the correlation dimension curve. Therefore, the intrinsic dimension estimation equals $a = 2.17461$, which is the intercept value in the linear equation of $y = 2.17461 + 3.06748\,(r)$.

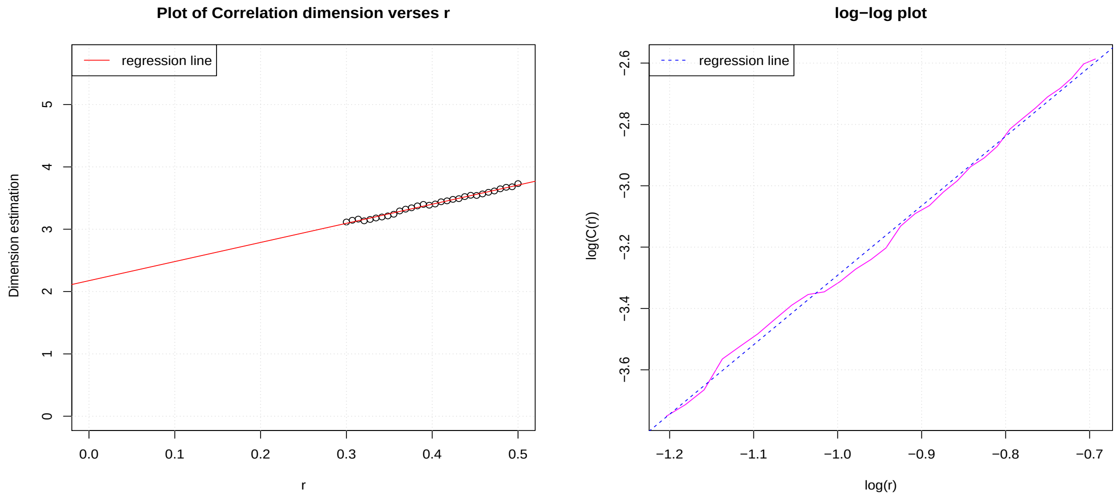
(b). Now, the implementation of the *Slope method*. Figure 5.3b displays the plotted curve of $\log(C(r))$ versus $\log(r)$ with a fitted linear regression. The estimated intrinsic dimension is equal to $b = 2.264904$. This value is close to the dimension value estimated by the Intercept method.

(c). Using the *Polynomial method*. We test the significance of parameters using a polynomial fit to $C(r)$ with degree 4. The results of the polynomial regression are



(a) (b)

Figure 5.2: Horse mussels data; (a) The fitted LPC- here is the plot of the *two*−dimensional pairwise projections onto the respective coordinate axes. (b) The ID estimations via Dip method.



(a) (b)

Figure 5.3: Horse mussels data; (a) Correlation dimension curve with a range of r from 0.3 to 0.5, (b) Log-log plot of correlation integral versus radius.

provided in Table 5.1. The most significant parameter is a_2, and hence, ID = 2, although the significance of a_1 is of similar magnitude, so there may also be evidence for ID = 1.

```
-----------------------------------------------------------------------
Coefficients:
        Estimate   Std. Error  t value  Pr(>|t|)
re      -0.07117    0.01076    -6.617   5.11e-07 ***
I(re^2)  0.57974    0.05712    10.150   1.55e-10 ***
I(re^3) -0.26064    0.09447    -2.759   0.0105   *
I(re^4) -0.02289    0.04897    -0.468   0.6440
---
Signif.codes:0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
Residual standard error: 0.001684 on 26 degrees of freedom
Multiple R-squared: 0.9998,     Adjusted R-squared: 0.9998
F-statistic: 4.026e+04 on 4 and 26 DF,  p-value: < 2.2e-16
-----------------------------------------------------------------------
```

Table 5.1: Horse mussels data; Summary table of the output of the Polynomial method.

Thirdly, the ID is estimated using a localized correlation integral method. The algorithm partitions the data set into several subregions. For this data the method constructs only three subregions. The ID is estimated for each subregion by applying the Intercept method and, as motivated in Section 4.4, the ID result is divided by 2. The ID estimate for the first subregion equals 0.5718384 with $n_1 = 27$, while for the second subregion the ID=1.2595886 with $n_2 = 27$ and the ID of third subregion is equal to 1.2225285 with $n_3 = 23$. Final, the median over all ID estimates equals 1.222528.

The next implementation is the MLE technique. We choose k between 10 and 20 where the algorithm presents reasonable ID estimates. Figure 5.4 shows the different estimations over the range of k, and the median ID is 2.504651.

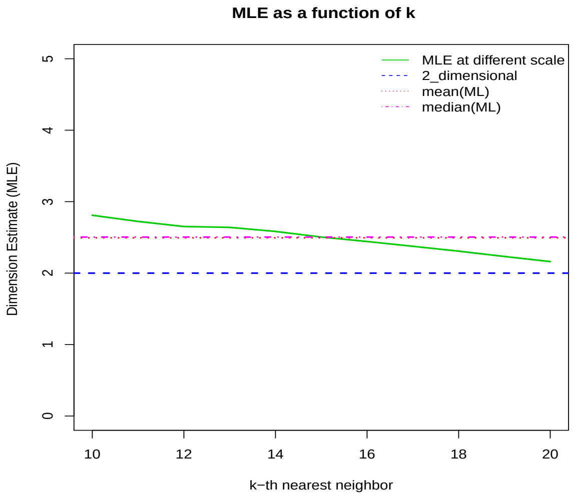


Figure 5.4: Horse mussels data; The dimensionality estimation via Maximum likelihood estimation.

We find that our approaches shows the dimensionality estimation of the Horse mussels data is 1 or 2, which is reasonable and matches the visual impression and scree–plot.

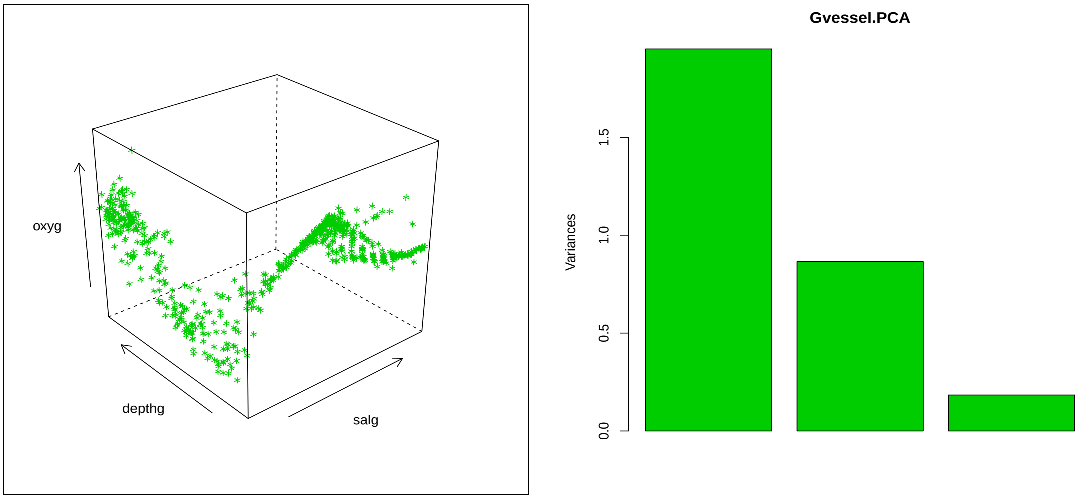
5.3.2 Oceanographic data

The Oceanographic data was collected by the German vessel, Gauss, in the North Atlantic, and retrieved from the World Ocean Database. The data is available in the package 'LPCM' under the names Gvessel data [25]. The data frame has 643 observations which were taken over nine days in May 2000. The Oceanographic data consists of seven variables, and for simplicity we will consider only three numeric measurements of variables salg, depthg and oxyg. The variables operate on different scales/units, salg is the ratio of electrical conductivity against a standard solution, due to the Practical Salinity Scale (PSS); depthg is the water depth in meters; and oxyg measures oxygen content in milliliters per liter of water. Figure 5.5a displays the scatter matrix of the three (scaled) variables.

A common starting point for the application is the scree plot as shown in Figure 5.5b. The three components of the PCA explain $65\%, 28\%$, and 6% of the total variance. One can conclude that the (linear) ID for this data set is about 2. However, closer inspection reveals that the cloud lies roughly on a curvilinear string through 3D space. Hence, we would intuitively expect its (nonlinear) ID not to be much larger than 1. This is plausible since linear ID estimates can be considered as an upper bound of their nonparametric counterparts.

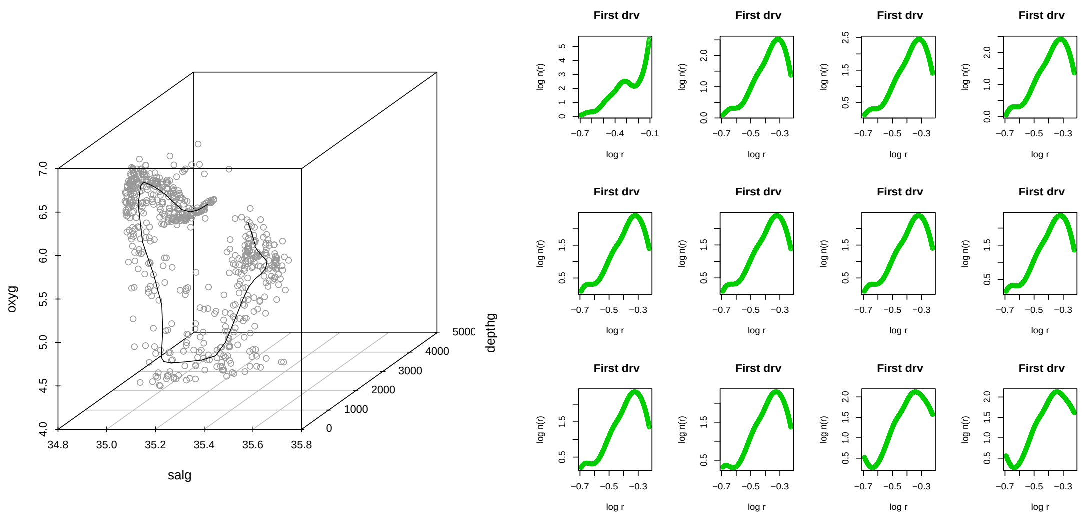
Firstly, the application of Brand's algorithm. As motivated earlier we fit the LPC to this data cloud, displayed in Figure 5.6a. An LPC is fitted through the data cloud with a starting point $x_0 = (35.7145, 48.39, 5.872)^T$ and bandwidth $h = 0.11$. The local centers of mass which define this curve are 'central' enough to avoid boundaries and provide good ID estimates. Hence, applying the above ID estimation routines on these local centers of mass, should on average, reproduce ID values which are close to 1. The ID is derived using the following methods.

(a). *Dip method.* Each curve in figure 5.6b represents the first derivative estimation for some selected LPC points. The median of all different ID estimations is 0.3088748. **(b)**. *Regression method.* Next we estimate the ID for each hyper–sphere of previous target points. By fitting linear regression to where the local ID is ob-



(a) (b)

Figure 5.5: (a) 3D scatter plot of scaled Oceanographic data, (b) Scree plot of linear PCA from scaled Oceanographic data.



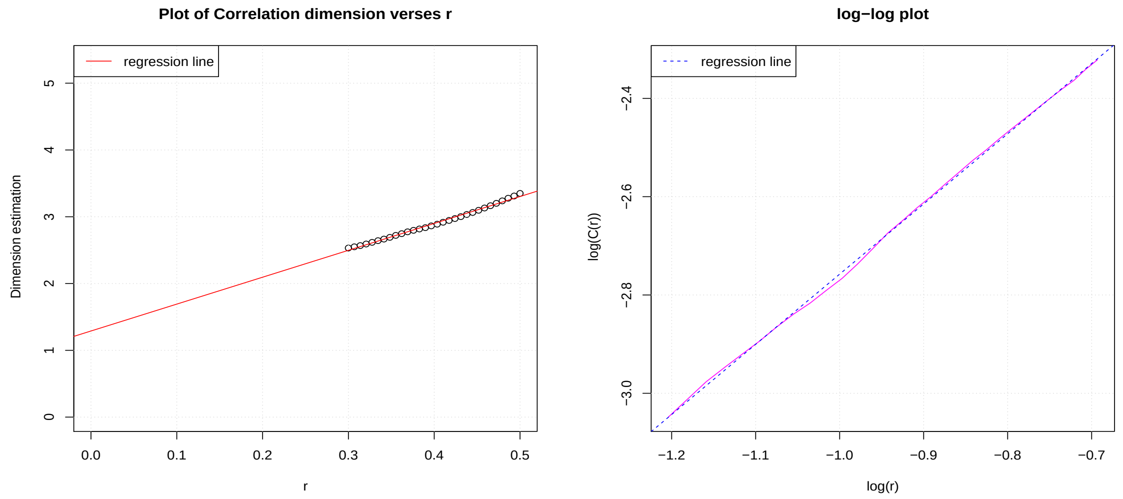
(a) (b)

Figure 5.6: Oceanographic data; (a) Fitting principal curve, (b) intrinsic dimension estimations at different target points (Brand's algorithm).

tained as the slope of the line, the ID is then derived by computing the median of the ID estimates which is equal to 1.423849.

Secondly, the implementation of ID estimation via the correlation dimension. **(a)**. *Intercept method*- We plot $D(r)$ versus versus r. Figure 5.7a shows that the curve looks to be reasonably linear from 0.3 to 0.5. Figure 5.7a displays the fitted regression line $D(r) = a + c\,r$ on the correlation dimension curve. Therefore ID $=$ $a =$ 1.289286, which is the intercept value in the linear equation of $y = 1.289286 +$ $4.027558\ (r)$. **(b)**. *Slope method*- The linear regression is fitted through the curve of $\log(C(r))$ in the loglog plot as shown in Figure 5.14b. As the linear equation is $y = -1.329790 +\ 1.427811 \log(r)$, the intrinsic dimension is equal to $b = 1.427811$. The result is reasonably close to the Intercept method. **(c)**. The implementation of the *Polynomial method*- We examine the parameter that has the largest t-value using a polynomial with degree 4. From the summary provided in Table 5.2 we immediately see that the largest parameter is a_2, and, hence, the estimated intrinsic dimension is equal to 2, i.e. ID $= 2$.

Next we consider the MLE implementation. For computational reasons, we take



(a) (b)

Figure 5.7: Oceanographic data; (a) $D(r)$ curve versus r, which is roughly linear for a reasonable range of r, (b) Log-log plot of correlation integral versus r.

```
-------------------------------------------------------------------------
Coefficients:
         Estimate  Std. Error  t value  Pr(>|t|)
re       0.031923   0.002515    12.70   1.19e-12 ***
I(re^2)  0.577557   0.013434    42.99   < 2e-16  ***
I(re^3) -0.603392   0.022371   -26.97   < 2e-16  ***
I(re^4)  0.211102   0.011660    18.11   2.93e-16 ***
---
Signif.codes:0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
Residual standard error: 0.0004065 on 26 degrees of freedom
Multiple R-squared:     1,      Adjusted R-squared:     1
F-statistic: 6.997e+05 on 4 and 26 DF,  p-value: < 2.2e-16
-------------------------------------------------------------------------
```

Table 5.2: Oceanographic data; Summary table of the output of Polynomial method.

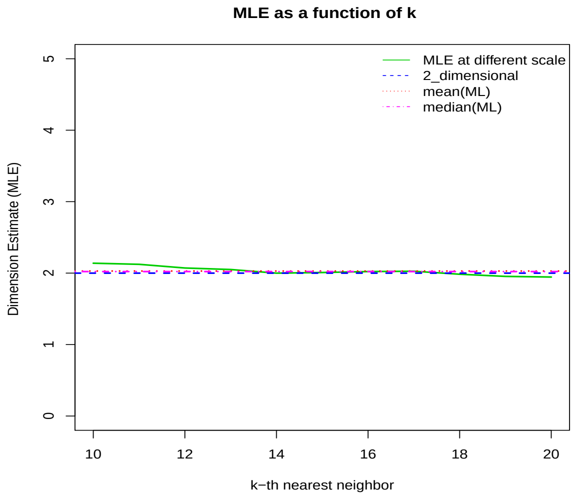


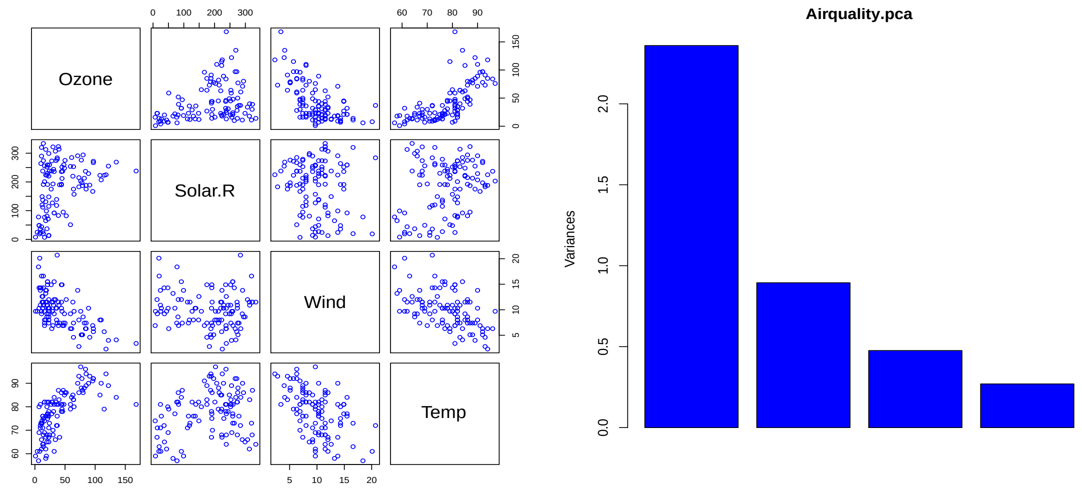
Figure 5.8: Oceanographic data; The dimensionality estimation via maximum likelihood estimation.

a sample of $n = 300$ data points and the range of k is $10 \ldots 20$, where for the very small value of $k_1 = 1$ the dimension estimator is -0.00762866. The resulting estimate is depicted in Figure 5.8, which shows different estimations over the range of k, and the final estimator 2.021688.

To sum up, the result of Dip method shows that the method fails for this data. In contrast the result from the MLE method agrees with the results of all other our approaches which are reasonable, and are consistent with the visual impression and the scree–plot.

5.3.3 Air Quality data

Air Quality data is based on the a daily measurement of air quality recorded in New York, during May to September 1973. Air quality data consists of numerical measurements of six variables: mean ozone(Ozone), solar radiation (Solar.R), average wind speed (Wind), maximum daily temperature (Temp), month, and day. We will only consider the first four measurements here. To gain an insight into the structure of the variables of the data, a pairwise plot of four-dimensional variable characteristics is provided in Figure 5.9a with 111 observations. In Figure 5.9b, the



(a)

(b)

Figure 5.9: Airquality data; (a) Pairwise plots, (b) Scree plot of four measurements of airquality data.

scree plot shows that three components explain 93% of the total variance of the scaled data, so depending on where one places the cut point, one would opt for IDs of 3 or 4. This result is intuitive when considering the data, which do not possess a very pronounced inner structure. Now, we compare these results to the estimated dimensionality via the Brand's and correlation dimension estimation methods.

Brand's algorithm - the intrinsic dimension is derived using the *Dip method* and the *Regression method*. Firstly, for the application of the Dip method, we consider the target points according to the highest–density–criterion outlined earlier. The first derivative estimator is derived using a local polynomial smoother with the bandwidth 0.15 for a sample of size 20 chosen from the higher density points as shown in Figure 5.10a. The median of all different ID estimations is 1.230157. Secondly, for the Regression method, we estimate the ID for each hyper–sphere of previous target points by fitting linear regression. The local ID is obtained as the slope of the regression line. Then, the ID is estimated by computing the median of the ID estimates which is equal to 1.638437. Both techniques provide results less than the result of MLE (Median ID=3.004193) at $k = 10, \ldots, 20$, as displayed in

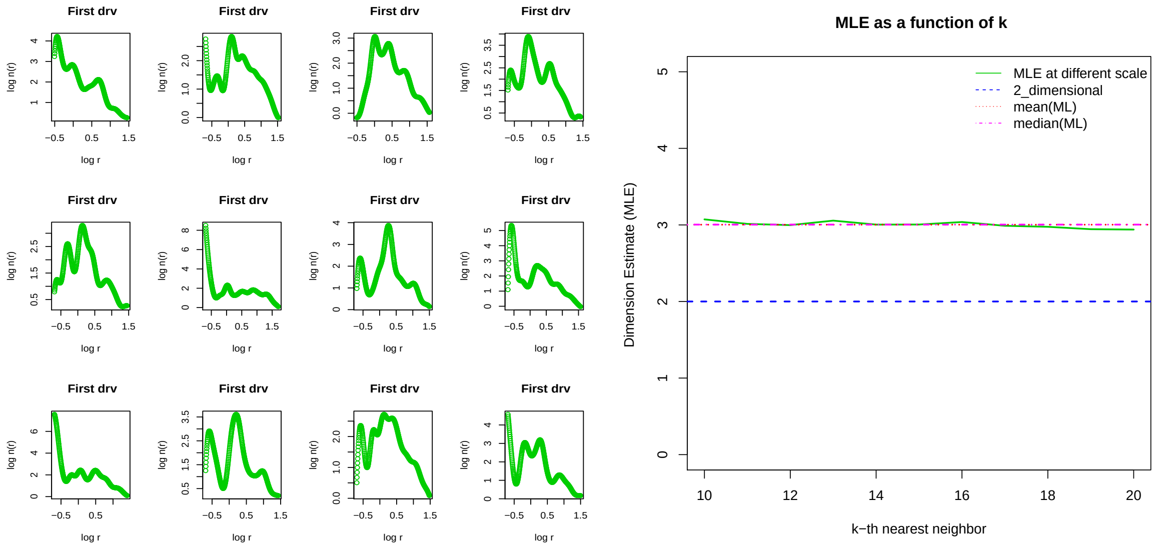
Figure 5.10b.

Next, the ID is obtained via the correlation dimension:

(a). *Intercept method*- We plot $c(r)$ versus versus r. Figure 5.11a shows that the curve of the correlation dimension is mostly linear in the chosen range of r. Figure 5.11a displays the fitted regression line $D(r) = a + c\,r$ on the correlation dimension curve. Therefore ID $= 3.438883$, which is the intercept value in the linear equation of $y = 3.438883 + 7.127591\ (r)$. **(b)**. *Slope Method*- The linear regression is fitted through the curve of $\log(C(r))$ in the loglog plot as shown in Figure 5.11b. The linear equation is $y = -2.279512 + 3.764282 \log(r)$, so the intrinsic dimension is equal to $b = 3.764282$. The result is reasonably close to the Intercept method.

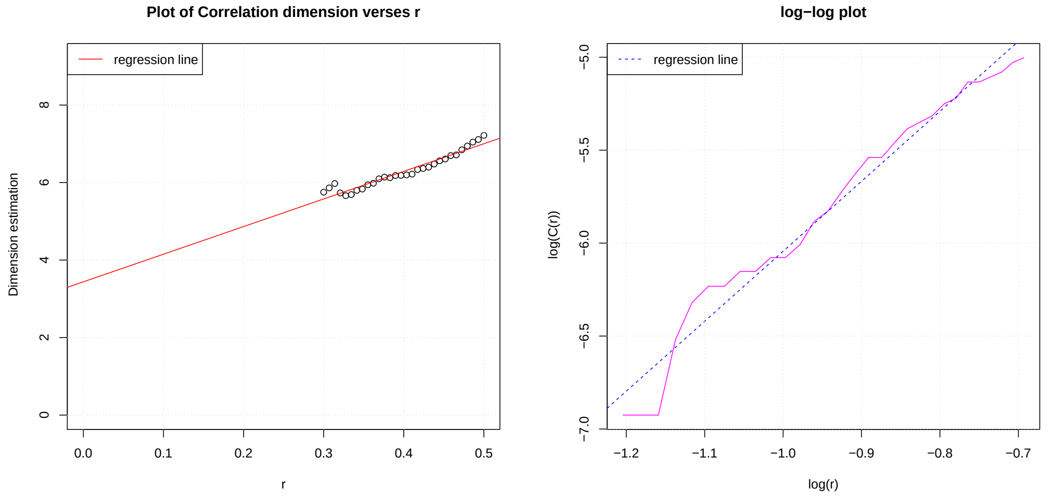
(c). *Polynomial method*- The ID is determined by investigating the significance of parameters using a polynomial with degree 4. From provided $*$ symbols in the summary (Table 5.3) we see immediately that the most significant parameter is a_3, and, hence, the estimated ID is equal to 3.

Thirdly, the ID is estimated using a localized correlation integral method. For this data the algorithm is iterated till the last subregion contains 18 data points. The method constructs four subregions and estimate ID locally using the Intercept



(a) (b)

Figure 5.10: Airquality data; (a) ID via dip method, (b) ID via MLE.



(a) (b)

Figure 5.11: Airquality data; (a) the plot of $D(r)$ versus r which is roughly linear for a reasonable range of r, (b) the log-log plot of correlation integral versus r.

```
--------------------------------------------------------------------------
Coefficients:
          Estimate  Std. Error  t value  Pr(>|t|)
re        0.0001382  0.0021715    0.064   0.949745
I(re^2) -0.0084240  0.0114175   -0.738   0.467237
I(re^3)  0.0771618  0.0187070    4.125   0.000337 ***
I(re^4) -0.0120179  0.0096240   -1.249   0.222886
---
Signif.codes:0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
Residual standard error: 0.0003192 on 26 degrees of freedom
Multiple R-squared: 0.9998,     Adjusted R-squared: 0.9998
F-statistic: 4.24e+04 on 4 and 26 DF,  p-value: < 2.2e-16
--------------------------------------------------------------------------
```

Table 5.3: Airqualty data: the result of fitting a polynomial of degree 4.

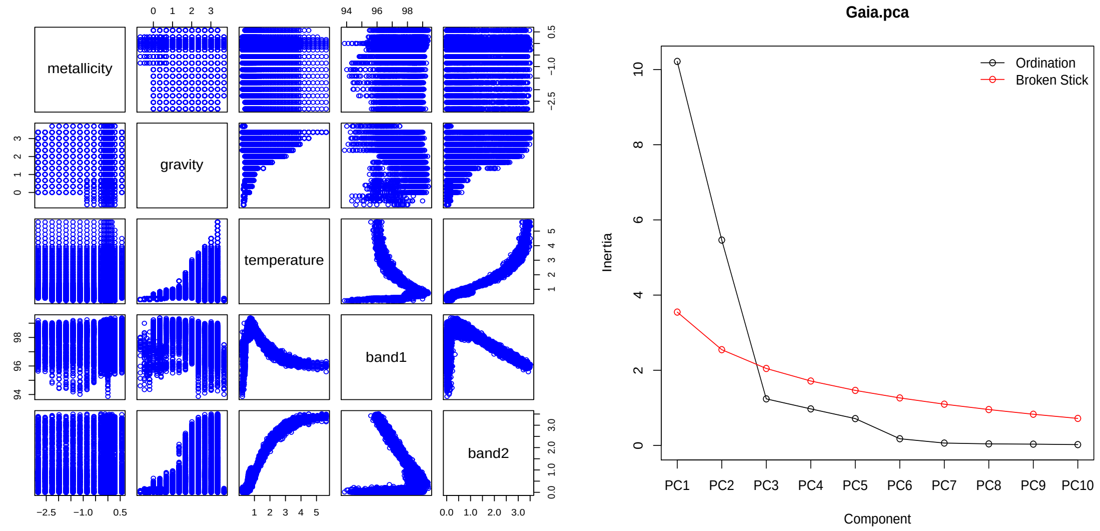
method. The ID result is divided by 2. Now, the ID estimate for the first subregion equals -0.7279828 with $n_1 = 22$, while for the second subregion the ID=2.4132869 with $n_2 = 39$ and the ID of third and fourth subregion is equal to 0.6577156 and 0.6912542 with $n_3 = 24$ and $n_4 = 18$, respectively. Final, the median over all ID estimates equals 0.6744849. In this case, we observe that the technique has correctly worked for the second subregion, for which the ID estimate is acceptable.

We find that the techniques (Intercept, Slope, Polynomial and MLE methods) arrive at sensible results which broadly agree with each other, and are consistent in line with the visual impression and the scree–plot. While the results using Dip method, Regression method and localized correlation integral provide underestimated ID.

5.3.4 Gaia data

Gaia is an European Space Agency (ESA) space observatory mission. It aims to collect data about the 1 billion stars in our Galaxy, and extragalactic objects. Gaia will provide comprehensive astrophysical information for each star, including its mass, temperature and chemical composition, among others. One of its major goals is to determine the distances, positions and annual proper motions of stars [2]. The data is available in the package 'LPCM' [25]. Gaia consists of two telescopes providing two observing directions with a fixed, wide angle between them. This samples the spectral energy distribution at 96 points across the optical and near-infrared wavelength range (3301000nm). The measurements themselves are photon counts (energy flux). Therefore each star can be represented as a point in a 96-dimensional data space.

We are going to analyze a simplified version of such data, which is generated by computer models. Our data set consists of photon counts measured in 16 (rather than 96) wavelength bands with 8286 observations. Additionally we include the three astrophysical parameters of temperature, metallicity, and gravity (which form the input space of the computer model) in our data set, giving a total of $D = 19$ dimensions for the raw data. For simplicity, Figure 5.12a displays the structure of only five variables of the data set. We begin our analysis by providing a scree plot in Figure 5.12b. The quickly falling curve starting in the left top provides the share



(a) (b)

Figure 5.12: Gaia data; (a) Scatter-plot matrix of five variables, (b) Scree plot of 19 variables.

of total variance explained by the respective principal component. The common way of interpreting this plot is by identifying sudden breakpoints, which separate the informative from the noise-carrying components. One finds here that there are two possible interpretations for this data set. There is a first break point at about 3 components, and a second (weaker) break point between 5 and 6 components. Alternatively, when performing linear dimension reduction via PCA, users can decide the dimension by how much variance they want to preserve. In the first case, 89% of the total variance is explained, while in the second case about 98% is explained. Note that the result $d = 3$ is backed up by the broken stick method, discussed in Subsection 3.4.1.

Now, we compare these results to the estimated dimensionality via Brand's and correlation dimension estimation methods.

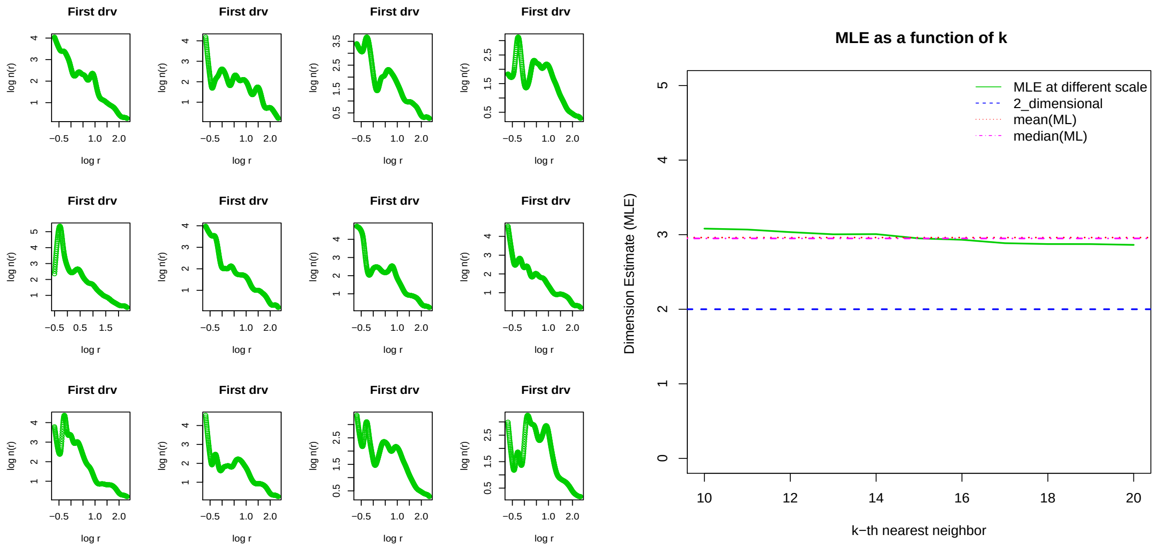
Firstly, the ID estimation via Brand's algorithm. We take a sample of 20 data points as target points according to the highest–density–criterion. Then:

(a). *Dip method.* The first derivative estimator is derived using a local polynomial smoother with bandwidth $h = 0.15$, as shown in Figure 5.13a. The median of all different ID estimations is 1.328673. **(b)**. *Regression method.* We estimate the ID for each hyper–sphere of previous target points by fitting linear regression and the local ID is obtained as the slope of the regression line. Then, the ID is estimated by computing the median of the ID estimates which is equal 1.515386. The ID results using both are underestimated comparing to the ID results using the MLE method (Median ID = 2.949778), as shown in Figure 5.13b.

Hence, the estimated dimensionality via the correlation dimension.

(a). *Intercept method.* We study the correlation dimension curve $D(r)$ as a function of radius r. As shown in Figure 5.14a, the curve of the correlation dimension looks to be reasonably linear in the chosen range of r. Figure 5.14a also displays the fitted regression line $D(r) = a + cr$ on the correlation dimension curve. Then the ID = 5.401008 which is the intercept value in the linear equation of $y = 5.401008 + 7.298104\ (r)$.

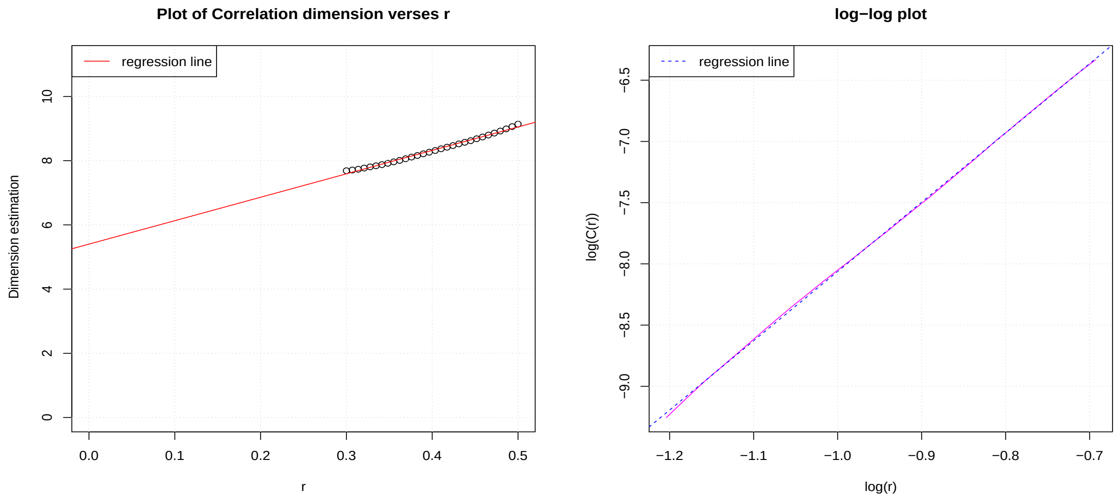
(b). *Slope method.* The plot in Figure 5.14b displays the curve of $\log(C(r))$ versus $\log(r)$ with a fitted linear regression. Therefore the estimated intrinsic dimension is equal $b = 5.657659$, this value is close to the dimension value estimated



(a)

(b)

Figure 5.13: Gaia data; (a) The implementation of Dip method, (b) ID via Regression method.



(a) (b)

Figure 5.14: Gaia data; (a) The implementation of Intercept method '$D(r)$ curve versus r', (b) Log-log plot of correlation integral versus r.

by the Intercept method.

(a). *Polynomial method.* The ID is derived by considering the largest t-value of parameters. For a polynomial of degree 4, one observes from Table 5.4 that the parameter with the largest t-value is a_3, so the intrinsic dimension of 3 is clearly identified.

We find that our approaches of implementation via correlation dimension indicate that the estimated intrinsic dimension for the Gaia data could be either at about 3 or at about 6, which are sensible results, and agree with the two possible interpretations from the PCA. Our variants of Brand's algorithm, *Dip method* and *Regression method*, representing a local ID estimation technique, produce an ID value of about less than 3 for this data and hence, favor the alternative PCA–based interpretation. In general, local methods will provide smaller IDs than global methods, since they are able to resolve the local data structure more flexibly [51].

It should also be noted that the results have a plausible physical interpretation. Since the input space is three-dimensional, and since the remaining 16 variables are generated from this input space, there is a strong argument for an intrinsic dimension

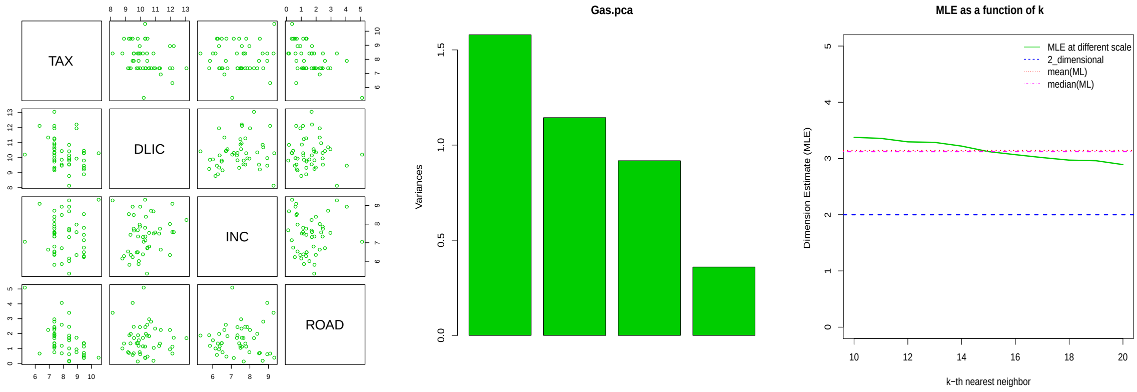
```
------------------------------------------------------------------------
Coefficients:
          Estimate  Std. Error  t value  Pr(>|t|)
re       0.0021728  0.0008447   2.572    0.0162   *
I(re^2) -0.0245780  0.0043751  -5.618    6.64e-06 ***
I(re^3)  0.0681822  0.0070749   9.637    4.55e-10 ***
I(re^4) -0.0262428  0.0036026  -7.284    9.79e-08 ***
---
Signif.codes:0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
Residual standard error: 0.0001133 on 26 degrees of freedom
Multiple R-squared: 0.9998,     Adjusted R-squared: 0.9998
F-statistic: 4.127e+04 on 4 and 26 DF,  p-value: < 2.2e-16
------------------------------------------------------------------------
```

Table 5.4: Gaia data; the result of fitting a polynomial of degree 4.

of 3. On the other hand, the 16-dimensional data cloud of photon counts, which has been simulated in some complex manner from the APs, will arguably increase the ID of the whole data set at least to some extent, where it is known that this increase should be less than three, since the first three principal component scores of the 16–dimensional photon counts are strongly correlated [23]. This is reflected in the ID of 5 obtained through the correlation dimension technique.

5.3.5 Fuel consumption data

Fuel consumption data consists of nine variables collected in $n = 48$ states of the United States of America. To determine the ID we consider only four continuous variables, these are TAX by cents per gallon, DLIC is the percentage of population who have driving licenses, INC the average income in \$1000s, and ROAD number of miles of road in thousands. Figure 5.15a displays a scatter-plot matrix, the scree plot of fuel consumption data and an example of point growth data at a random sample point above the density threshold. It clearly shows that the data has no specific structure with moderate noise. Here the three components explain 91% of



(a) (b) (c)

Figure 5.15: Fuel consumption data; (a) Scatter-plot matrix, (b) Scree plot, (c) ID via MLE.

the total variance of the scaled data, therefore, depending on where one places the cut off point one would decide that the (linear) ID is about 3. Figure 5.15c illustrates the ID estimation via MLE $= 3.122107$, for range of k between 10 and 20, where at $k = 1$ we get -0.009722454.

Firstly, the implementation via Brand's algorithm. It is obvious that due to $n = 48$, the number of highest–density data points is small, less than 20. We take a sample of 10 data points as target points according to the highest–density–criterion. The result of the implementation is provided in Table 5.5 below.

Variants of Brand's algorithm	ID
Dip method	1.210457
Regression method	1.821634

Table 5.5: The estimated IDs via Brand's algorithm.

Secondly, the implementation via correlation dimension. In practice, for this data, the requirement of having at least two data points within the sphere leads to a minimum r equal to 0.34, then the range of r between 0.34 and 0.5. In addition, for the Polynomial method we take the maximum point of r equals 2. At this point the result looks reasonable. The results of the three techniques are provided in Table 5.6 below.

Variants of correlation dimension	ID
Intercept method	3.518778
Slope method	4.072675
Polynomial method	3

Table 5.6: The estimated IDs via correlation dimension algorithms.

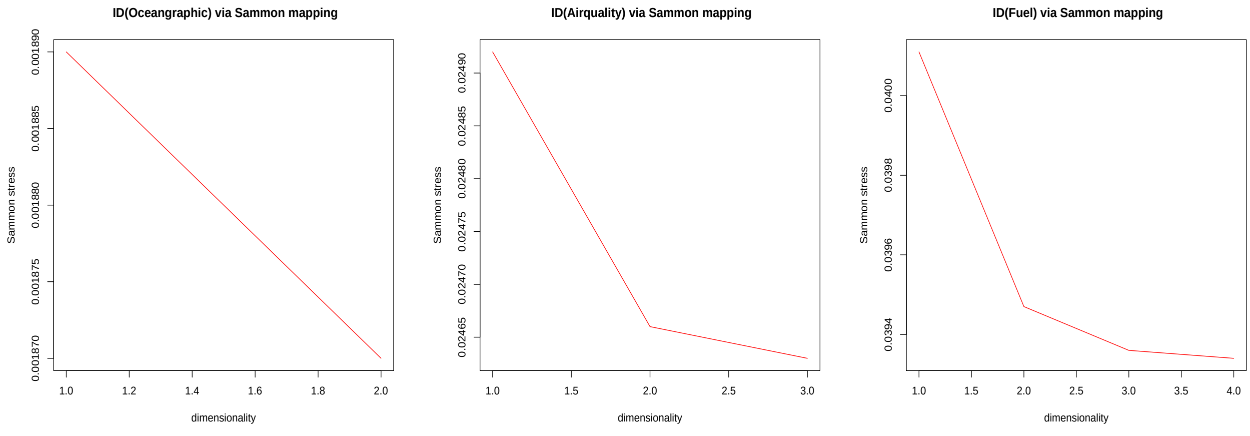
5.4 Comparisons

In this section we provide comparative experimental results on the data set which compare the scree plot (global linear ID method), MLE (local nonlinear ID methods) and MDS method (global nonlinear ID methods) regarding to our approaches. The results are summarized in Table 5.7.

	Data set				
Method	Horse mussels	Oceanographic	Air qual.	Gaia	Fuel cons.
D	4	3	4	19	4
N	82	643	111	8286	48
Dip	0.82	1.41	1.5	1.33	1.21
Regression	1.55	1.19	1.64	1.52	1.82
Intercept	2.17	1.29	3.44	5.4	3.52
Slope	2.26	1.43	3.76	5.66	4.07
Polynomial	2	2	3	5 or 6	3
MLE	2.50	2.02	3.00	2.95	3.12
Scree Plot	≈ 1	≈ 2 or 3	≈ 3 or 4	≈ 3 or $5-6$	≈ 3
MDS	–	–	2	–	2

Table 5.7: The estimated IDs for several data sets (where Air qual.: Air quality data and Fuel cons.: Fuel consumption data).

Figure 5.16 illustrates the ID estimates via multidimensional scaling method (MDS) for Oceangraphic, Airquality and Fuel consumption data. It is important to note that the MDS method usually projects data points onto a two-dimensional manifold, which means that it is assumed that the ID = 2 in the algorithm. We apply the MDS algorithm and the intrinsic dimensionality is obtained by plotting the minimum stress versus the dimensionality. Then the ID value is shown as a knee or a flatting of the curve (see Subsection 3.4.2). For Oceanographic data, Figure 5.16a indicates that the knee does not exist to obtain ID, which is the drawback of the MDS algorithm, see Subsection 3.4.2. Similar unsatisfactory results using the MDS method were obtained for the Gaia and the Horse mussels data (graphs not



(a) (b) (c)

Figure 5.16: The ID estimate via MDS method; (a) Oceanographic data(b) Air quality data, (c) Fuel consumption.

shown).

From the results in Table 5.7, we make three observations. First, we find that our techniques (Dip, Regression, Intercept, Slope and Polynomial methods) arrive at sensible results which, apart from a few exceptions, broadly agree with each other. The performance for our approaches on the expermintal data sets is the same compared to the performance of these methods on artificial data sets. Second, the results of the implementation via correlation dimension are consistent with the visual impression and the scree plot, which tends to suggest slightly larger IDs. In contrast, our variants of Brand's method, local methods, provide a reasonable but possibly underestimated ID estimate since they estimate the ID of the subregion of the data set. Third, we observe that the implementation via correlation dimension is faster than using variants of Brand's methods.

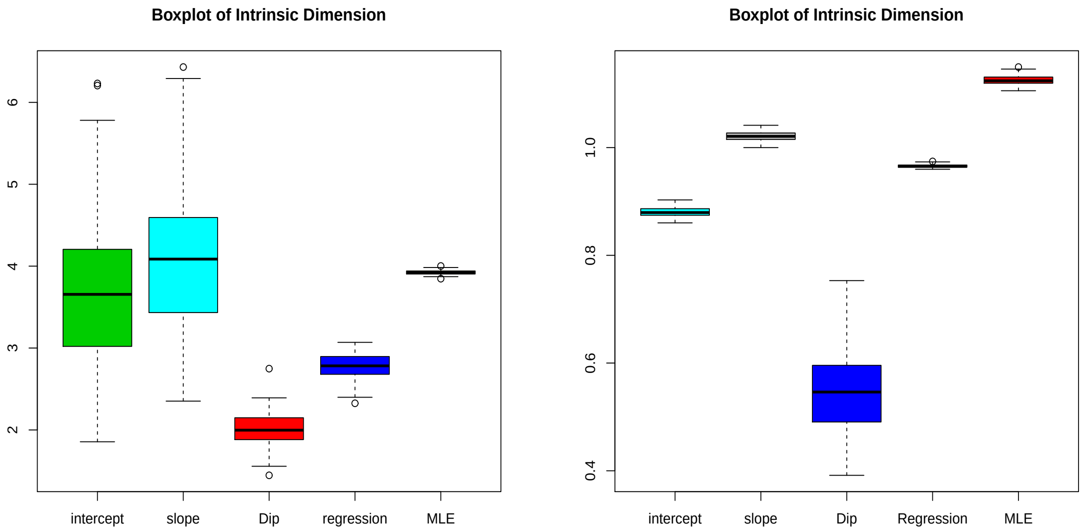
To sum up, our methods estimate ID using the geometric properties of the data, and do not require the parameters to be set. The Experimental results on both artificial data, as shown in Section 4.6, and real data illustrate that our approaches enable us to estimate ID. In the next section we provide a simulation study in the next section which will be more conclusive in terms of the actual performance of the methods.

5.5 Simulation studies

The purpose of this section is to present the precision of our approaches. We generate data sets of known ID and try to identify their ID through MLE method, Brand's method and correlation dimension by considering three cases for $d = 1, 2$ and 4. As illustration we provide box-plots which show the median and distribution of ID estimates via the MLE method, the Dip method, the Regression method, the Intercept method and the Slope method, while the results for the Polynomial method will be presented in tabular form.

Firstly (a), a data set of size $n = 200$ with dimension $D = 4$ is generated from a multivariate Gaussian distribution with parameters $m = (9, 5, 6, 4)$, where the diagonal of the covariance matrix Σ is equal to $(50, 50, 50, 50)$. Since this data do not possess any inner structure, we would assume the ID to be equal (or close to) 4 in this case. We generate 100 data sets in this manner, and for each sample we calculate the ID estimate. The result in Figure 5.17a indicates that the methods provide reasonable ID estimates. In fact the slope method overshoots slightly with a median slope estimate of 4.086142, while the median of the IDs obtained via the Intercept method is 3.655468, and the median of the IDs using MLE is equal 3.919751. Also the Figure 5.17a shows that the ID via the Regression method is equal (median = 2.782664), while the median using Dip method is equal 1.997086, perhaps indeed a bit underestimated. Note that the polynomial method provides an integer ID, rather than continuous values. From Table 5.8 we see that both the median and mode of the estimated ID values take the value 4, closely followed by an estimated dimension of 3, which confirms the results of the other techniques nicely.

Secondly (b), the data is generated by adding four–variate Gaussian noise e to data distributed uniformly on a straight line (think of a long cigar–like object in 4D space), with zero mean vectors and unit covariance matrices such that $E\left(ee^T\right) = 0.0025I$. We would assume this data has an ID roughly equal to 1. Again, we provide a box-plot of the ID estimates for 100 simulated data sets. The plot in Figure 5.17b illustrates that our methods obtain a good estimate of the intrinsic dimension, with the Slope method achieving results (median: 1.020941) which are closer to 1 than



(a) (b)

Figure 5.17: Simulation study; box plot of ID estimates via Intercept, Slope, Dip, Regression and MLE methods of 100 data sets generated from multivariate Gaussian distribution. (a) First case, (b) Second case.

the Intercept method (median: 0.8794059), while the median of the IDs via MLE is 1.124016. Comparing those methods with the other local approaches via the Dip method and the Regression method, the median ID using the Regression method is 0.9652246, which is closer to 1, while the median using the Dip method is 0.5461858, which means it has underestimated ID. The Table 5.8 shows that the Polynomial method returned the correct ID of 1 throughout.

Thirdly (c), we use the simulation setup that was provided by Liu et al. [62]. Consider a process of five–variate Gaussian noise z is constructed as a linear combination of $s = (s_1, s_2, s_3)$ such that:

$s_1(i) = 2\cos(0.08\,i)\ \sin(0.06\,i)$,

$s_2(i) = \text{sign}[\sin(0.03\,i) + (9\cos(0.01\,i))]$,

$s_3(i) \sim N(0, 0.25)$.

where i is a sampling index. Assume the process is $x = y + e$ and $y = Bs$ that is, a model of type 3.1, where

$$B = \begin{bmatrix} 0.860 & 0.790 & 0.670 \\ -0.550 & 0.650 & 0.460 \\ 0.170 & 0.320 & -0.280 \\ -0.330 & 0.120 & 0.270 \\ 0.890 & -0.970 & -0.740 \end{bmatrix},$$

$E\left(ee^T\right) = 0.0025I$, and $E(e) = 0$. Now we should suppose that the data have ID $= 3$ or smaller. A total of 100 samples were simulated from that process. Figure 5.18 displays the box-plot of the ID estimates (Intercept, Slope and MLE), the result provides reasonable ID estimates, with the Intercept method achieving results (median$= 2.440109$) which are close to the Slope method (median$= 2.60165$), whereas the median ID estimated via MLE is 3.919751. While the results via the Dip method and the Regression method present an underestimated ID, the median is $0.9651418, 1.031972$, respectively. As seen from Table 5.8, the polynomial method has returned ID $= 2$, in 100% of the simulation runs.

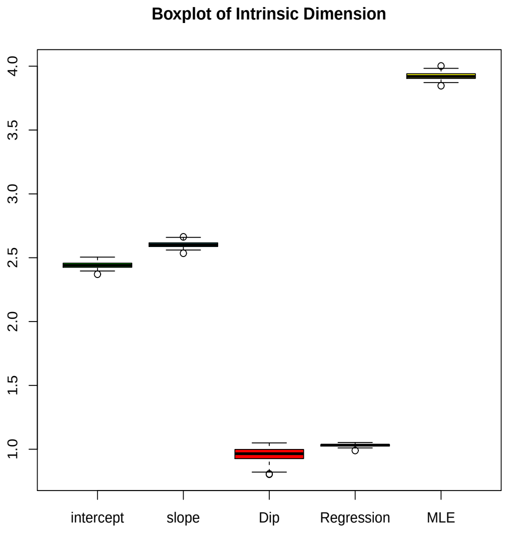


Figure 5.18: Simulation study; The box plot of ID estimates via Intercept, Slope, Dip, Regression and MLE methods of 100 data sets generated from multivariate Gaussian distribution for the third case.

	estimated ID			
Sim.	1	2	3	4
a)	0	2	47	**51**
b)	**100**	0	0	0
c)	0	**100**	0	0

Table 5.8: Summary of results of polynomial method for simulations (a), (b), and (c). The bold number shows the proportion of 'correctly' identified intrinsic dimensions.

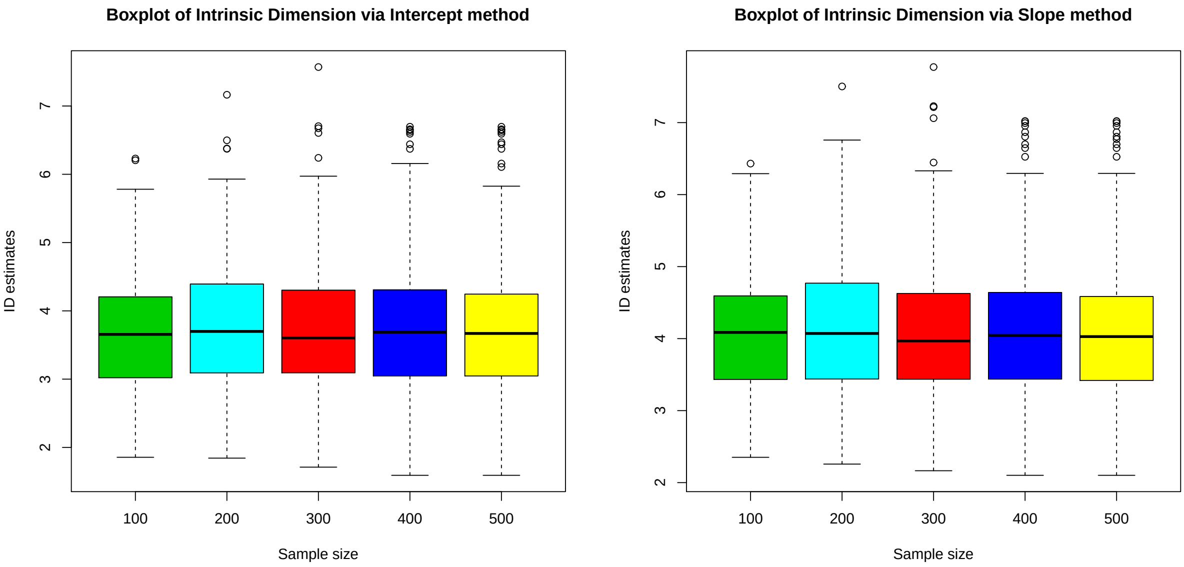
5.5.1 Discussion of Bias and sample size

In this section, in terms of the performance of the methods, we discuss the bias of the estimators and the effect of the sample size on the accuracy of the ID estimates using simulation data. Firstly, the effect of bias. From results in Figures 5.17a, 5.17b and 5.18, we observe that the Slope method provides an overestimated ID for the first case, while it has the smallest bias for the second cases. Though the Intercept method appears slightly negatively biased, its ID estimates of mostly ≤ 4 are more plausible than those of the slope method. Even though both Intercept and Slope methods have a larger variance for the first case. For the third case of simulation, both Intercept and Slope methods achieve similar results, of plausible magnitude, about 2.5. Compared to other approaches the Dip method has the the largest bias and the Regression method is the next largest bias. Note that the under–estimation is a feature of the local methods. In contrast, MLE has the largest bias for the third case, and it is known that its bias increases with high dimension because it needs a very large data sample in the sphere [60]. These results are consistent with the discussion of results on artificial data (Section 4.6.3).

Secondly, the effect of the sample size N. For simplicity and computation time, we will discuss the performance with first case (case (a)) and second case (case (b)). We provide box plots of ID estimates for different simulation sample sizes for both cases, as shown in Figures 5.19, 5.20, 5.21, 5.22 and 5.23. In addition, the median is computed over all ID estimates for each sample size, and the results for both cases are displayed in Tables 5.9 and 5.10. The results in the figures and tables show that our approaches do not depend on sample sizes and the results do not differ much for each sample size. As we mentioned in Subsection 4.6.3, the local methods provide an under–estimation (a lower bound) ID, since it needs to estimate the ID for each sphere. In contrast, the global ID methods provide an estimate greater than the estimate provided by local methods.

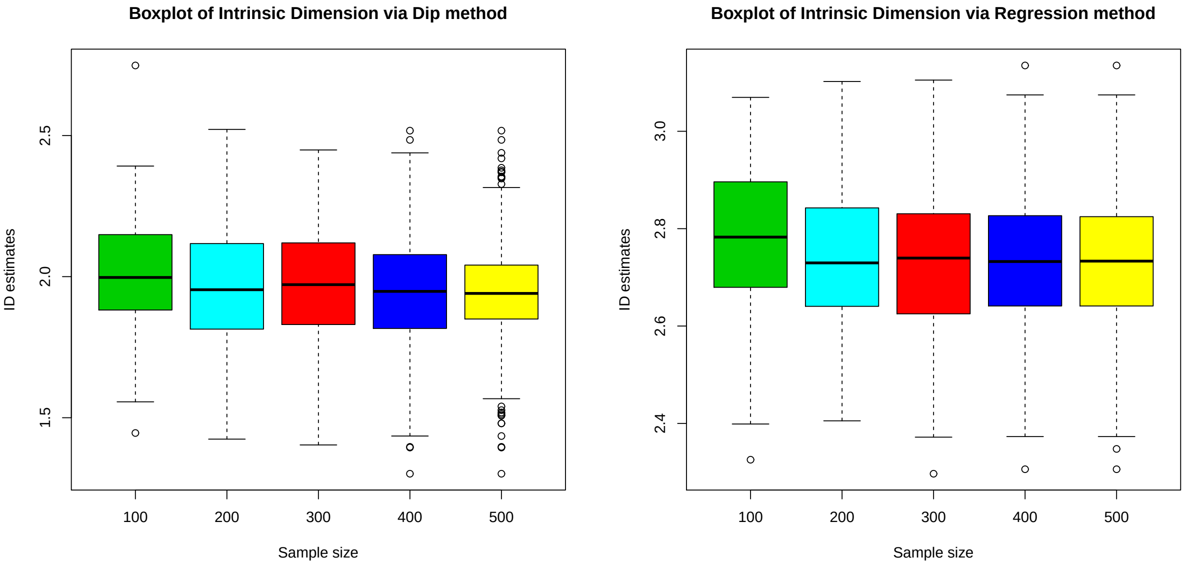
In particular, our experimental results establish that the main weakness of local techniques for dimensionality estimation is the requirement to estimate the ID at several subregions which leads to increased computation time. We may suggest a technique that does not rely on the data's local properties. It has been suggested that the dimensionality estimation could be obtained by applying global ID methods at subregions, as shown in Section 4.4. The main value of local ID methods for dimensionality estimation is that they can be applied on data sets where we do not have enough information about the global structure available, such as Melter data (Section 5.6).

In this piece of research comparisons between variants of Brand's algorithms (Dip and Regression methods) and the correlation dimension (Intercept, Slope and Polynomial methods) show that the Intercept and the Slope methods behave similarly, and consistently give ID estimates which are closer to the real ID than other methods. Furthermore, the results of the experiments carried out on the previous data sets seem to suggest the same conclusion.



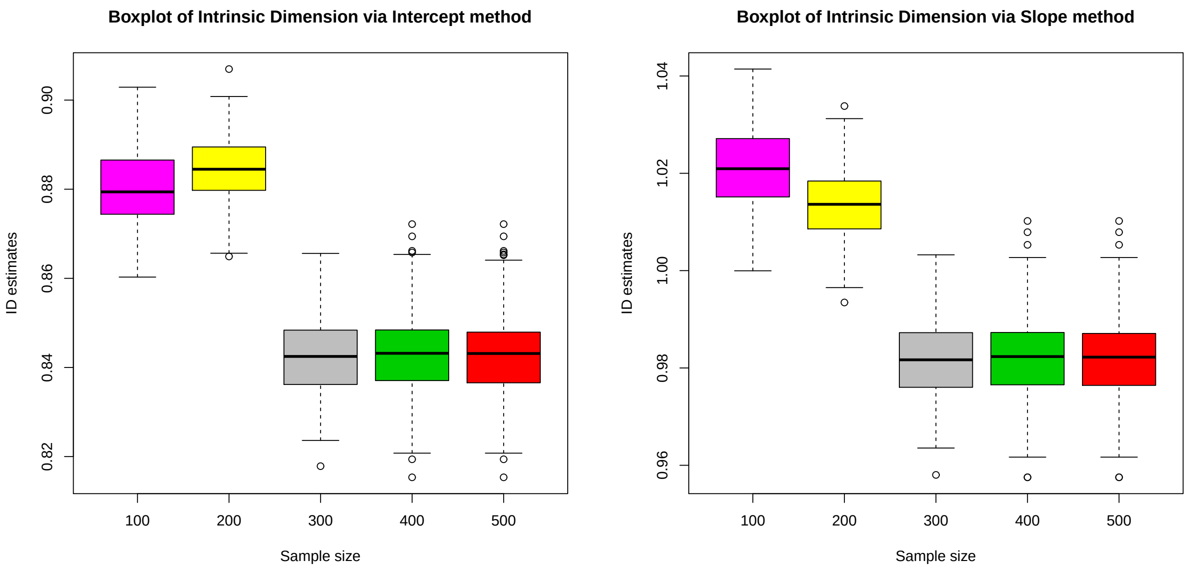
(a) (b)

Figure 5.19: Box plots of ID estimates via our approaches of different sample sizes for the first case: (a) Using Intercept method, (b) Using Slope methods.



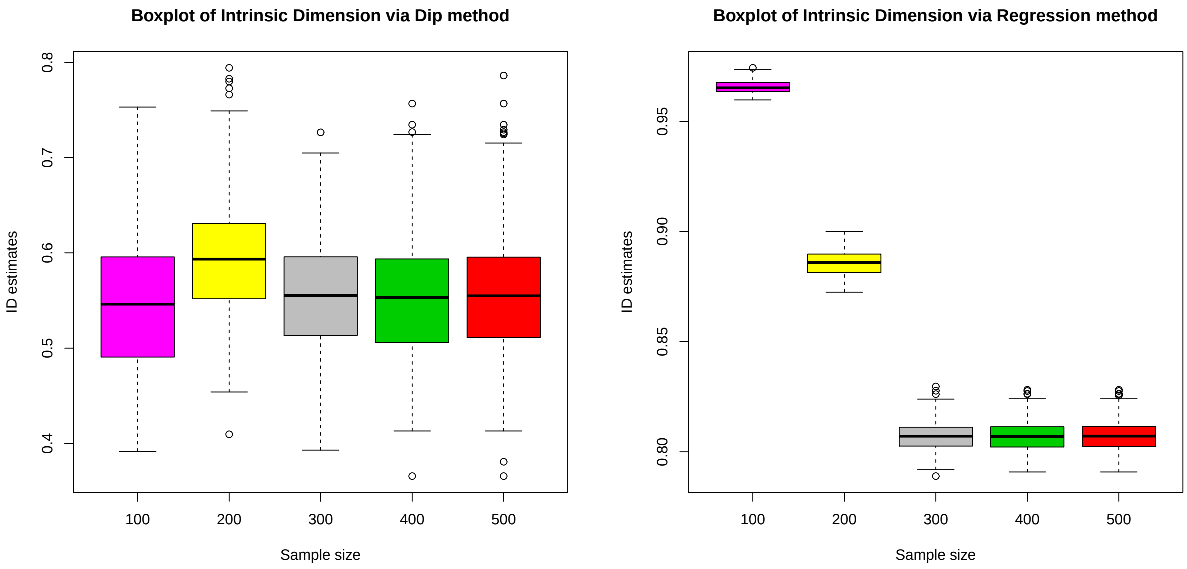
(a) (b)

Figure 5.20: Box plots of ID estimates via our approaches of different sample sizes for the first case: (a) Using Dip method, (b) Using Regression methods.



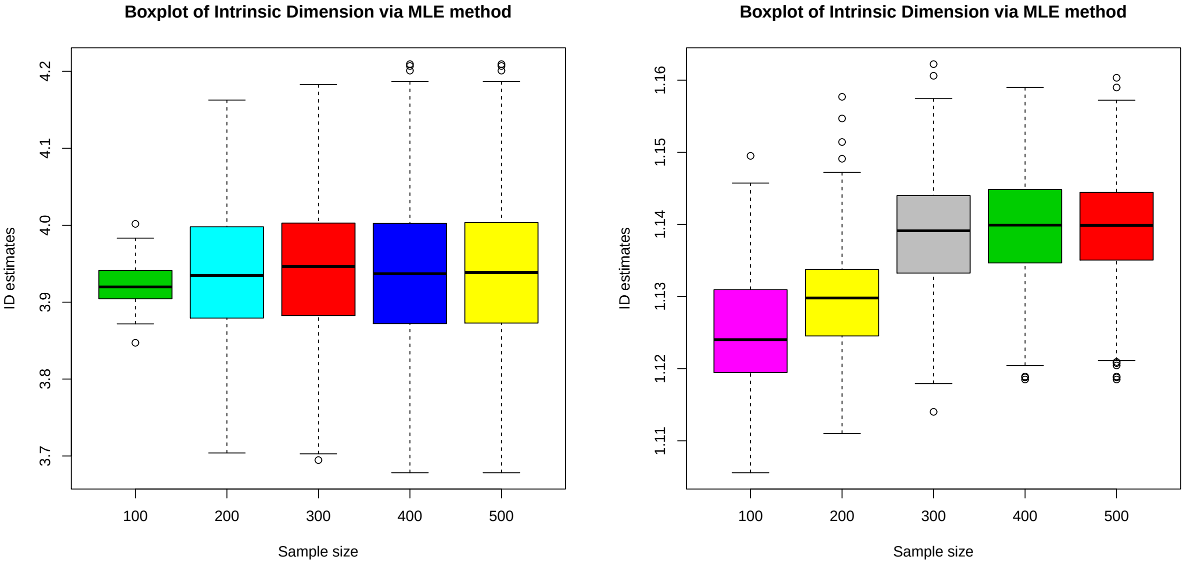
(a) (b)

Figure 5.21: Box plots of ID estimates via our approaches of different sample sizes for the second case: (a) Using Intercept method, (b) Using Slope methods.



(a) (b)

Figure 5.22: Box plots of ID estimates via our approaches of different sample sizes for the second case: (a) Using Dip method, (b) Using the Regression methods.



(a) (b)

Figure 5.23: Box plots of ID estimates via MLE method with different sample sizes; (a) the first case, (b) the second case.

	Methods					
N	Intercept	Slope	Polynomial	Dip	Regression	MLE
100	3.66	4.09	4	1.99	2.78	3.92
200	3.70	4.07	4	1.95	2.73	3.93
300	3.60	3.97	4	1.97	2.74	3.95
400	3.69	4.04	4	1.95	2.73	3.94
500	3.67	3.78	4	1.94	2.73	3.90

Table 5.9: The median(ID) over all ID estimates via our approaches at several sample size for the first case (true ID=4).

5.6 Melter data

The Melter data are industrial data measured within a glass melter at high temperatures. The data consists of 21 variables with $N = 17280$ data points. The variables are: the measurements of fifteen temperature sensors, the electric power

	Methods					
Sample size	Intercept	Slope	Polynomial	Dip	Regression	MLE
100	0.88	1.02	1	0.55	0.97	1.12
200	0.88	1.01	1	0.59	0.89	1.13
300	0.84	0.98	1	0.55	0.81	1.14
400	0.84	0.98	1	0.55	0.81	1.14
500	0.84	0.98	1	0.55	0.81	1.14

Table 5.10: The median(ID) overall ID estimates via our approaches at several sample sizes for the second case (true ID=1).

measurements of four induction coils, the viscosity of the molten glass, and the electric voltage. We are going to analyze this data by only considering a sample of $n = 2000$ data points. For simplicity, Figure 5.24 displays a scatter plot of only 12 variables of Melter data.

We establish our analysis by providing a scree plot in Figure 5.25a. The quickly falling curve starting in the left top provides the share of total variance explained by the respective principal component. One can infer here that there is a break point at about two components. With two components, 88% of the total variance is explained, while with four components 95% is explained. Note that the result $d = 2$ is backed up by the broken stick method. Now, for simplicity we take a subsample of $n' = 300$ data points and the ID is derived locally using MLE method. The plot in Figure 5.25b displays ID estimation at the selected k from 10 to 20, and, the median of all ID estimates equals 4.662585.

We now compare these results to the estimated dimensionality via Brand's algorithm and correlation dimension.

Intrinsic dimension estimation obtained using Brand's method. Firstly, for the *Dip method*, we choose a sample of size 50 of target points according to the highest–density–criterion outlined earlier. The ID is estimated for each target point by computing the first derivative with bandwidth 0.15 as shown in Figure 5.26. The

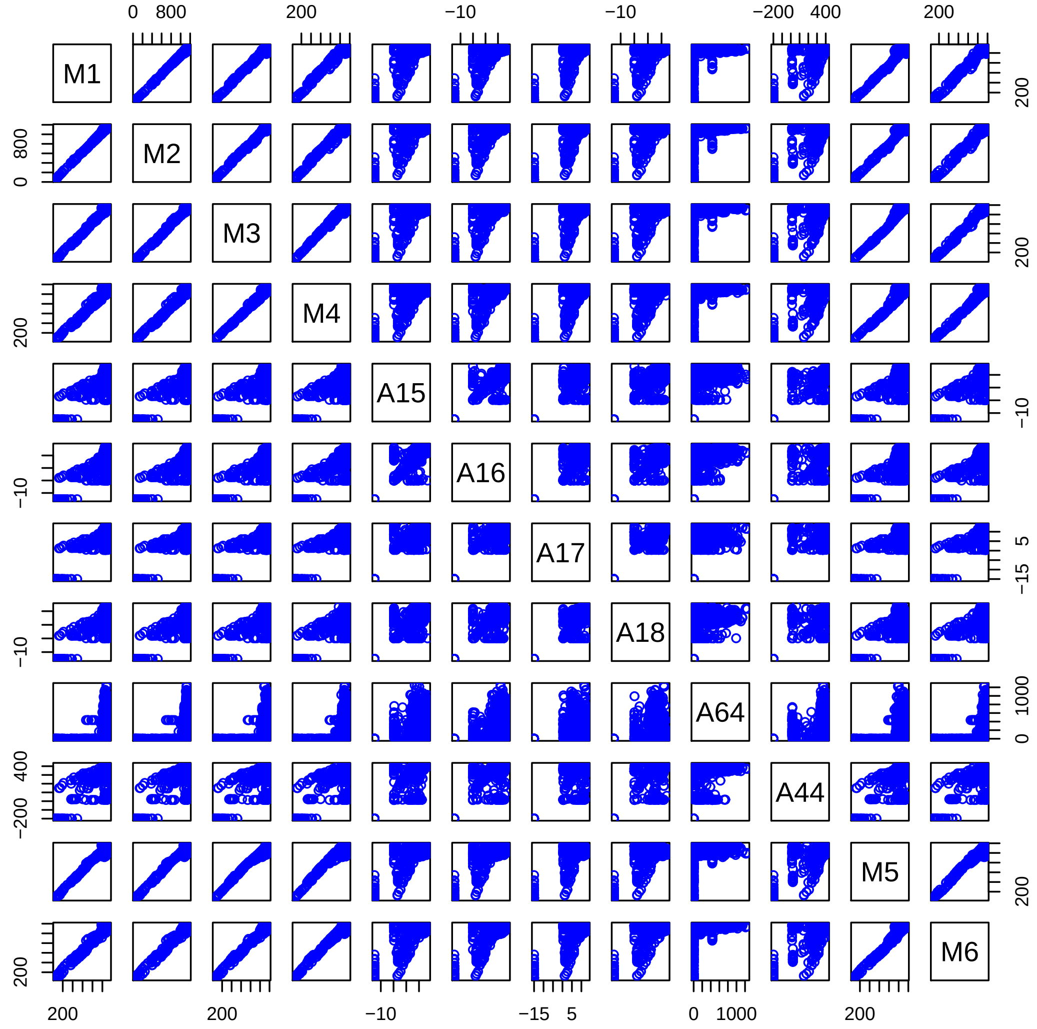
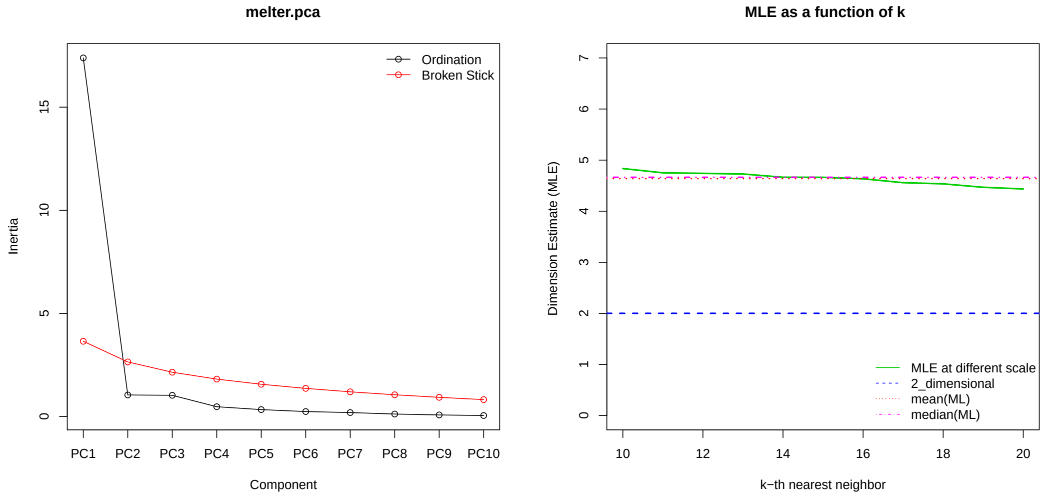


Figure 5.24: Scatter-plot matrix of 12 variables of Melter data.

median of all different ID estimations is 2.900821. Secondly, the *Regression method*, the ID is estimated by fitting linear regression method on the previous target points and determining the slope of the regression. Then the ID is derived as the median of the ID estimates which is equal 1.1548.

Next, the dimensionality is estimated via correlation dimension. Firstly, *Intercept method* implementation. We study the correlation dimension curve with radius r. Here, as shown in Figure 5.27a the curve of correlation dimension looks to be reasonably linear in the chosen range of r. Figure(5.27a) displays the fitted regression



(a) (b)

Figure 5.25: (a) Scree plot of Melter data with 21 measurements, (b) The dimensionality estimation of Melter data via MLE method.

line $D(r) = a + cr$ on the correlation dimension curve. Then, ID = 1.483556 which is the intercept value in the linear equation of $y = 1.483556 + 14.102666\ (r)$. Secondly, the testing of the *Slope method*. The plot in Figure 5.27b displays the curve of $\log(C(r))$ versus $\log(r)$ with a fitted linear regression. Therefore, the estimated intrinsic dimension is equal to $b = 1.913968$. This value is close to the dimension value estimated by the Intercept method.

Finally, using the *polynomial method*. The ID is derived via a series of $t-$ tests on the model parameters. We assume that the correlation integral is modelled by a polynomial of degree 4. The results are shown in Tables 5.11 and 5.12 with the upper value of r equal to 1 and another trial with equal to 1.5, respectively. From Table 5.11, the most significant parameter is a_1, and hence, ID = 1. Now, with upper point of r equals 1.5, Table 5.12, we choose the parameter that has the largest t-value rather than the smallest p-value, since the p-values are too small to be distinguished. Therefore a_3 has the largest t value, so may also be evidence for ID = 3. In addition, for polynomial of degree 7 with the upper value of r equal to 1.7, the parameters a_1 and a_4 provide similar magnitude, one can infer that ID

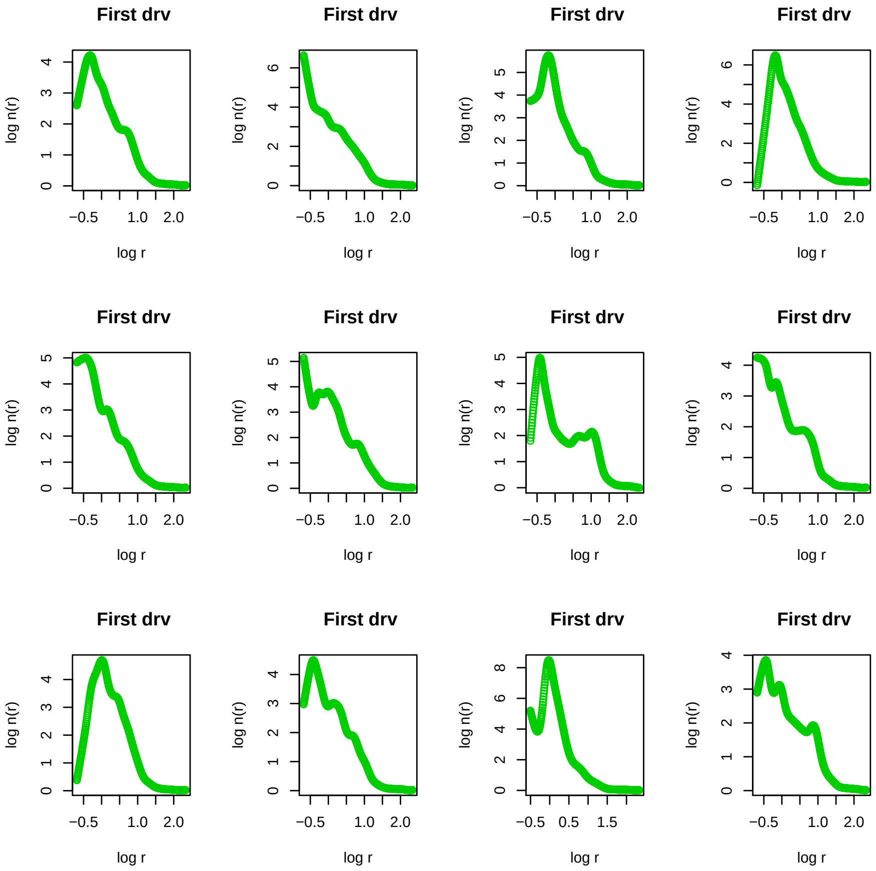
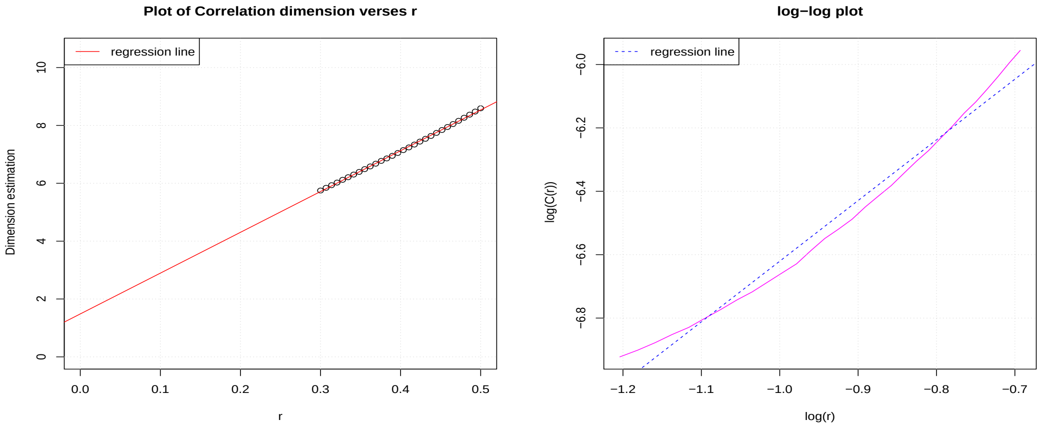


Figure 5.26: Melter data; The estimated dimension via Dip method

could be either 1 or 4 as displayed in Table 5.13.

We find that there is some discrepancy in the observed dimension estimates. While the intuitive scree-plot based solution of $ID \approx 2$ is backed up by the dip method and the correlation-based techniques, we observe a larger value of ≈ 4.7 via the ML method, and a smaller value of ≈ 1 for the regression method and the polynomial method. It appears that the latter, very small, ID estimates are possibly flawed and the polynomial method tends to be especially fragile for large data dimension. As far as the correlation methods are concerned, Camastra and Vinciarelli [9] observe that, for small sample sizes, the correlation integral tends to underestimate the true ID (in this context, 2000 is still a 'small' sample size), and provide a 'reference curve' which is supposed to remove the downwards bias. However, for very small correlation dimensions (such as 1 or 2) this concept appears unsuitable (since the reference curve would deliver an ID of 0 in this case). The MLE solution of 4.7 appears overestimated given the evidence provided by all other techniques.

Motivated by the results of this section, we attempted to model the melter data through a 2-dimensional principal manifold. We used the experimental R package 'lpmforge' (Evers, 2013) [29], which implements the extension of the local principal curve method illustrated in Section 2.3.2 to 'local principal manifolds'. In the special case $d = 2$, the manifold is a 'surface'. The resulting surface is displayed in Figure 5.28 that shows that the assumption of ID=2 appears plausible. The ID may actually be higher in the denser part, which could explain the different results of the ID estimation.



(a)

(b)

Figure 5.27: Melter data; (a) The correlation dimension curve with range of r, (b) The log-log plot of correlation integral versus r.

```
-------------------------------------------------------------------------
Coefficients:
          Estimate Std. Error t value Pr(>|t|)
re       0.0099929  0.0009256  10.796 4.20e-11 ***
I(re^2) -0.0405625  0.0049430  -8.206 1.09e-08 ***
I(re^3)  0.0592029  0.0082274   7.196 1.21e-07 ***
I(re^4)  0.0040489  0.0042861   0.945    0.354
---
Signif.codes:0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
Residual standard error: 0.0001472 on 26 degrees of freedom
Multiple R-squared: 0.9999,     Adjusted R-squared: 0.9998
F-statistic: 4.824e+04 on 4 and 26 DF,  p-value: < 2.2e-16.
-------------------------------------------------------------------------
```

Table 5.11: Melter data; At upper point of radius equals 1 the result of fitting a polynomial of degree 4.

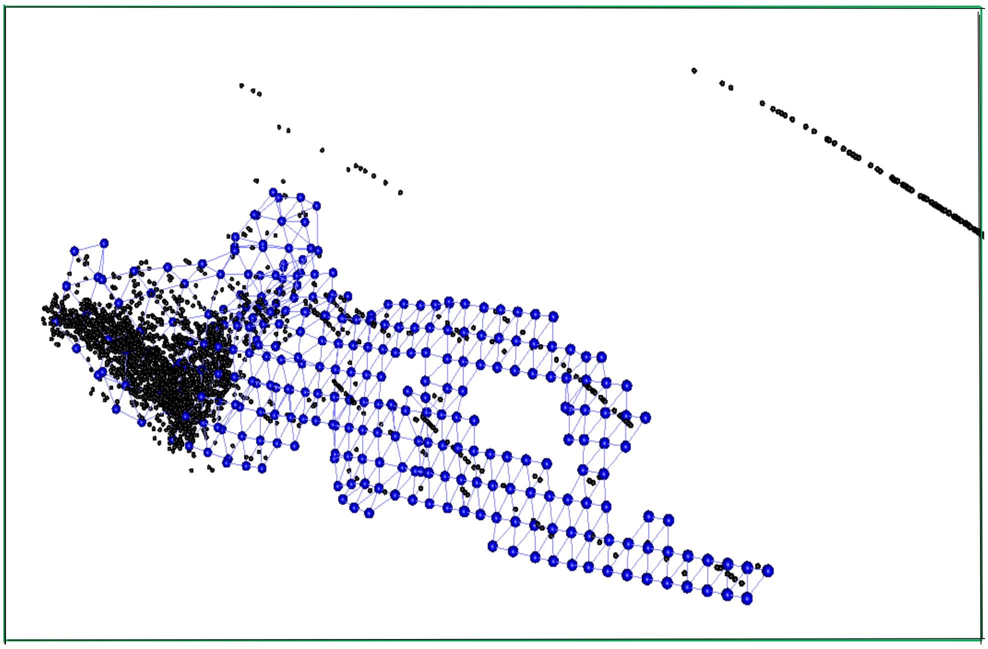


Figure 5.28: Melter data; The principal manifold implementation

```
-----------------------------------------------------------------------
Coefficients:
          Estimate Std. Error t value Pr(>|t|)
re       0.0135155  0.0008451   15.99  5.7e-15 ***
I(re^2) -0.0677897  0.0030088  -22.53  < 2e-16 ***
I(re^3)  0.1111882  0.0033387   33.30  < 2e-16 ***
I(re^4) -0.0243939  0.0011595  -21.04  < 2e-16 ***
---
Signif.codes:0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
Residual standard error: 0.0002015 on 26 degrees of freedom
Multiple R-squared:     1,      Adjusted R-squared:     1
F-statistic: 3.99e+05 on 4 and 26 DF,  p-value: < 2.2e-16.
-----------------------------------------------------------------------
```

Table 5.12: Melter data; the result of fitting a polynomial of degree 4 with upper point of radius equals 1.5.

5.7 Conclusion

In this chapter, we assessed the effectiveness of the proposed algorithms in the light of real data examples. A simulation study was also provided. A comparison was made with the PCA method, MLE and MDS methods. PCA provides an upper bound dimension, that is, the value of the dimension is often larger than in nonparametric ID estimation methods.

In contrast, local ID methods provide a lower bound of ID since they estimate the ID of subregion of the data set. The localization also leads to increased computation time.

In practice, with the variants of Brand's algorithm, it is noted that besides the choice of the target point, the range of the sequence of radius and the length of the grid of the radius value could effect the graph and the estimation of ID. However, all those factors do not seriously affect the ID estimation.

Regarding our approaches to the computation of the correlation dimension method, it is clear that the chosen range of r is influenced by the part of the correlation di-

```
------------------------------------------------------------------------
Coefficients:
         Estimate Std. Error t value Pr(>|t|)
re        0.003348   0.001890   1.771   0.0898 .
I(re^2)   0.013330   0.019298   0.691   0.4966
I(re^3)  -0.113297   0.073874  -1.534   0.1388
I(re^4)   0.261343   0.137319   1.903   0.0696 .
I(re^5)  -0.176941   0.132921  -1.331   0.1962
I(re^6)   0.048531   0.064405   0.754   0.4588
I(re^7)  -0.003823   0.012335  -0.310   0.7594
---
Signif.codes:0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
Residual standard error: 0.0001063 on 23 degrees of freedom
Multiple R-squared:     1,      Adjusted R-squared:     1
F-statistic: 8.189e+05 on 7 and 23 DF,  p-value: < 2.2e-16
------------------------------------------------------------------------
```

Table 5.13: Melter data; the result of fitting a polynomial of degree 7.

mension curve that looks linear. Our methods reduce the computation time since we consider the small r that is our main interest, and avoid counting pairs with large r.

The correlation dimension occupies the middle ground between purely linear methods (such as PCA) and purely topological methods (which average over localized IDs representing the dimension of the tangent space along the manifold). Indeed, the IDs obtained via the correlation dimension are generally equal to or smaller than the ID suggested by a scree plot (broken stick, etc $\cdots$), and larger than the estimates obtained through local (topological) techniques, such as Brand's (2003) algorithm.

For our approaches, the comparisons between variants of Brand's algorithms (Dip and Regression methods) and the correlation dimension (Intercept, Slope and Polynomial methods) show that the Intercept and the Slope methods behave similarly, and consistently give ID estimates which are closer to the real ID than other method. Furthermore, the results of the experiments carried out on the previous data sets seem to suggest the same conclusion.

For our implementation of the Localized correlation integral method, we have no actual justification that the ID needs to be divided by 2, even if our considerations at the beginning of Section 4.4 makes this a plausible thing to do. The results are based on an experimental implementation of the localized correlation integral method, and further research would be necessary to investigate whether the results do indeed give reliable ID estimates.

Chapter 6

Discussion and Future work

6.1 Summary of the Thesis

Dimension reduction is a key concept in many real-life statistical applications such as data mining and pattern recognition. Most dimension reduction methods require an explicit definition of the intrinsic dimension (ID) of the low-dimensional subspace, as shown in Section 2.4. Additionally, we illustrated the relationship between intrinsic dimension and some dimension reduction methods in Section 2.4. As an example, in order to fit the principal curve to the Spiral data, within a two-dimensional space, the user firstly has to decide that the ID is equal to 1, as displayed in Figure 2.5. This means that the intrinsic dimension should be fixed in advance before applying dimension reduction methods. There have been few attempts dedicated to determining how to estimate the ID of data in this context. This thesis develops methods on the basis of the existing concepts.

Firstly, a brief review of dimension reduction methods was given in Chapter 1. Dimension reduction methods can be categorized as linear or nonlinear methods. Linear methods try to search a globally flat subspace, such as principal component analysis. Nonlinear methods try to search a locally flat subspace, such as multidimensional scaling methods and ISOMAP. Several dimension reduction methods are related to each other. For instance linear PCA is a special case of the kernel PCA with a linear kernel, ISOMAP is a special case of MDS by using geodesic distances, and MDS is a special case using ISOMAP with k (the number of nearest neighbors) equal to $N - 1$. The computation cost is influenced by the number of parameters of the technique and the number of iterations required. Most nonlinear methods have parameters which need to be optimized, for instance techniques that are based on neighbors such as ISOMAP and LLE. Although nonlinear methods have higher computational costs than linear methods, these costs are offset by improvements to the performance of sub-sequential statistical inference.

The estimation of intrinsic dimension is very useful for dealing with real-life data with high dimensions, such as a data set $Z = \{x_1, \ldots, x_N\} \in \mathbb{R}^D$ which we assume to be scaled, i.e. each variable has been divided by its standard deviation. When the intrinsic dimension of Z is given as a value d, this gives effectively the minimum number of variables necessary to describe the data without much loss of information [8] [32]. ID estimation methods can be classified into two groups: *local* methods which divide the data into small subregions and estimate the ID in each subregion, and *global* methods which try to estimate the dimension using the whole data set. An overview of methods of intrinsic dimension estimation was presented in Chapter 2. The global method is widely used in the manner of PCA. Projection methods and MDS are used as dimension reduction methods rather than dimensionality estimation methods.

On the other hand, local and global ID methods suffer from a bias of high dimension, where the bias appears to be due to inadequate sampling. This occurs when the sample is from the region near the edges or boundaries of a manifold [54]. It is noted that the correlation dimension has the smallest bias and the MLE has the next smallest bias [60]. All methods also require large samples in high–dimensions which could increase the computational cost.

From the implementation on the artificial data, we observe that techniques (such as PCA and Nonlinear PCA, that do not employ neighborhood graphs) provide unreasonable ID results. The bias in the PCA method appears due to the linearity constraint. Regarding the Kernel PCA, the bias comes from the specific nonlinearity constraint imposed, which is influenced by the kernel function and the parameter changes of the function. These methods implemented previously are basically used as dimension reduction methods.

In addition, we observe that MLE method gives a visual impression of positive bias, but is consistent with the scree-plot (linear PCA). The bias in the MLE method occurs because the neighborhoods need to contain sufficient data points, which is difficult for a finite sample size.

As far as we know, although nonlinear methods, global or local methods are available, it appears that not enough work had been done on implementing the methodology of the estimation of dimensionality on non-linear manifolds. Furthermore, as with many methods, there is not enough evidence that they work well practically. For instance, Charting a manifold method needs to identify target points. Also fractal methods require the construction of a correlation integral, from which the ID is extracted using appropriate techniques. This step is not straightforward, since it requires counting the number of data pairs within a ball of radius tending to 0.

The objective of this thesis has been to provide new approaches for the calculation of ID via Brand's algorithm and correlation dimension. We have proposed algorithms which are versions of Brand's algorithm, and the ID is obtained locally via Dip and Regression methods. The Intercept, Slope and Polynomial methods estimate the ID globally via correlation dimension. All these methods could be classified as nonparametric methods, as opposed to linear methods such as PCA. Conceptually, the 'linear' intrinsic dimension should provide an upper bound for IDs achieved via nonlinear methods, and in fact, we observed that the values suggested by PCA-based ID are often larger than those obtained by nonparametric ID estimation methods. To be even more precise, within the nonparametric methods, we found that global methods tend to produce larger IDs than local methods.

The correlation dimension occupies some middle ground between purely linear methods (such as PCA) and purely topological methods (which average over localized IDs representing the dimension of the tangent space along the manifold). Indeed, the IDs obtained via the correlation dimension are generally equal to or smaller than those suggested in a scree plot (broken stick, etc $\cdots$), but larger than the estimates obtained through local (topological) techniques, such as Brand's algorithm. In addition, we have also estimated the ID by computing the median of

Maximum likelihood estimates for a data set. A discussion of the practical implementation of the methods (artificial data sets, experimental data sets and simulation data) is given in Chapters 4 and 5.

The concepts introduced in this thesis are not restricted to a particular type of application. We have given different examples – from the environmental and physical sciences – where the methods were clearly useful. They could also be applied to data sets of any kind, including, for instance, data (bases) which are created and collected on the internet.

For the Dip and Regression methods, it is clear that not only the choice of target points, but also the starting point of the sequence of radius and the length of the grid of radius values could impact upon the graph and the estimation of ID.

For the approaches via correlation dimension we have investigated three techniques, two of which are novel, to implement fractal ID estimation via the correlation integral. Both the Intercept and Slope methods provide non-integer ID estimates, while the Polynomial method provides an integer value. The Polynomial method is novel and appealing, but difficult to use for data sets in high dimension D, because a polynomial degree $d \leq p \leq D$ needs to be chosen. The proposed techniques, the Intercept and Slope methods, require relatively few data points and are not demanding on the sample size. Examples with real data verify the concept of estimating correlation dimension at exactly $r = 0$.

For the Intercept and Slope method, the chosen range of r is motivated by the part of the respective curve that looks approximately linear. These regions of linearity may differ between different data sets, but we have provided default choices, which, according to our experience, work well for a wide range of data sets.

For the Polynomial method, we have to be close to 0, but not too close, as we need more data, because we are fitting a more complex model. Hence we need a larger upper r as compared to these other methods. The Polynomial method is of theoretical appeal and the result needs to be extracted manually from the regression output.

In particular, our experimental results establish that the main weakness of local techniques for dimensionality estimation is the requirement to estimate the ID at several subregions, which leads to increased computation time. We suggest a technique that does not rely on the data's local properties. It has been suggested that the dimensionality estimation could be obtained by applying global ID methods at subregions, such as the proposed method Localized correlation integral. The main value of local ID methods for dimensionality estimation is that they can be applied on data sets where we do not have enough information about the global structure available, such as the Gaia and Melter data sets.

The Localized correlation integral is proposed by implementing the Intercept method locally on disconnected subregions of data sets. In our implementation of the Localized correlation integral method, we have no actual justification that the ID needs to be divided by 2. However our considerations at the beginning of Section 4.4 justify that this a plausible thing to do. The results are based on the experimental implementation of the localized correlation integral method, and further research would be necessary to investigate whether the results do indeed give reliable ID estimates.

In summary, a simulation study has confirmed that the Intercept, Slope and Polynomial methods provide ID estimates which, on average, are close to the underlying 'true' ID. Additionally, the Intercept and Slope methods are compared and shown to behave similarly, and consistently give ID estimates which are closer to the real ID than other method. Furthermore, the results of the experiments carried out on the previous data sets seem to suggest the same conclusion. However, the ID estimate via the Dip method underestimates the ID and the Regression method also tends a little bit to underestimate the ID. In addition, the simulation study indicates that the MLE is biased when applied to high-dimension data set. It must be noted that all this is non-causal. The value d may underestimate the number of variables needed for applications such as regression.

The overall conclusion reached is that all the methods we have proposed in this thesis are easy to implement and apply, and the experimental analysis indicates that these methods are suitable for dealing with various types of data, including linear and non–linear structures. Our own code for the implementation of our new approach is available in http://www.maths.dur.ac.uk/∼dma0je/zakiah.

Chapters 4 and 5 are my original research. Section 4.2 was presented in the ISM conference [51] while Section 4.3 was discussed at the ICSSBE2012 conference [50] and published in [22]. A further manuscript [28] is in preparation. Each of these four papers use selected examples presented in Chapter 5.

6.2 Suggestion for future research

The process of developing this thesis has led me understand that there are other ways of taking the research forward and building upon it. The following points summarize several possible areas for investigation in the future:

1. Explore other ways to estimate the ID by applying nonlinear global methods locally on subregions, and then obtain the ID for the data set by averaging over all ID estimates.

2. Investigate other suitable ways to select the target point of Brand's algorithm. For example, by taking the points that are close to the mode or the median of the data set.

3. Exploring further experimental implementation on the Charting with pairs approach in order to investigate whether the results do indeed provide robust and reliable ID estimates, and then compare these to the experimental results of the localized correlation integral.

4. Exploring whether a nonlinear correlation coefficient could be useful for nonlinear ID estimation.

5. Investigate ID estimation when the focus is not on unsupervised learning, but on a particular inferential problem, such as regression. The question to ask would be: what is the 'best ID' to use to predict a certain response variable?

Appendix A

Math

A.1 Abbreviations and Symbols Used

X: A D-variate random vector.

$g(x)$: Probability density distribution.

$\hat{g}(x)$: Kernel density estimator of $g(x)$.

Ω: $\Omega = \{x_1, \ldots, x_N\} \in \mathbb{R}^D$ is a sample of size N is drawn from the random vector X.

Σ: Variance covariance matrix.

$\hat{\Sigma}_{ML}$: Maximum likelihood estimator of Σ.

$\hat{\Sigma}_{sample}$: Sample variance matrix estimator of Σ.

N: Sample size.

n: Subsample size.

ID: Intrinsic dimension.

PCA: Principal component analysis.

$G(r)$: Brand's expression.

MLE: Maximum Likelihood Estimator.

d_H: Hausdorff dimension.

d_{box}: Box-counting fractal dimension.

d_{cor}: Correlation dimension.

$C(r)$: Correlation integral which is the proportion of distance pairs.

$N(r)$: Number of data points in sphere of radius r.

$H(r)$: The inverse function of $G(r)$.

$d_k(x)$: MLE for dimension d.

ICA: Independent component analysis.

LDA: Linear discriminant analysis.

PV: Principal variables.

ANN: Autoassociative neural network.

PC: Principal Curve.

LPC: Local Principal Curve.

LPM: Local Principal manifold.

MDS: Multidimensional Scaling.

ISOMAP: Isometric feature mapping method.

LLE: Locally Linear Embedding method.

SOM: Self-Organising Maps.

ViSOM: Visualisation induced SOM.

TRN: Topology representing network.

A.2 Proof of the result in Section 4.3.1

Assume that $C(r)$ is a polynomial with degree $p \geq 1$. Hence, let

$C(r) = a_p r^p + a_{p-1} r^{p-1} + \ldots + a_3 r^3 + a_2 r^2 + a_1 r + a_0$.

Considering the condition $C(0) = 0$, we get $a_0 = 0$. Then $C(r)$ can be written as

$$C(r) = a_p r^p + a_{p-1} r^{p-1} + \ldots + a_3 r^3 + a_2 r^2 + a_1 r.$$

The estimate of d via correlation dimension, according to Eq. (4.6) where $d_{cor} = d$, becomes:

$$d_{cor} = \lim_{r \to 0} \frac{\log\left(a_p r^p + \ldots + a_3 r^3 + a_2 r^2 + a_1 r\right)}{\log(r)}.$$

Next, applying l'Hopital's rule we get:

$$\begin{aligned} d_{cor} &= \lim_{r \to 0} \frac{r\left(p a_p r^{p-1} + \ldots + 3 a_3 r^2 + 2 a_2 r + a_1\right)}{a_p r^p + \ldots + a_3 r^3 + a_2 r^2 + a_1 r} \\ &= \lim_{r \to 0} \frac{p a_p r^p + \ldots + 3 a_3 r^3 + 2 a_2 r^2 + a_1 r}{a_p r^p + \ldots + a_3 r^3 + a_2 r^2 + a_1 r}. \end{aligned}$$

Applying l'Hopital's rule a second time we get:

$$d_{cor} = \lim_{r \to 0} \frac{p^2 a_p r^{p-1} + \ldots + 9 a_3 r^2 + 4 a_2 r + a_1}{p a_p r^{p-1} + \ldots + 3 a_3 r^2 + 2 a_2 r + a_1}.$$

at $r \to 0$, then

$$d_{cor} = \frac{a_1}{a_1} = 1.$$

Now, suppose that $a_1 = 0$ and $a_0 = 0$, then

$C(r) = a_p r^p + \ldots + a_3 r^3 + a_2 r^2$.

Then, substitute to d_{cor} gives:

$$d_{cor} = \lim_{r \to 0} \frac{\log\left(a_p r^p + \ldots + a_3 r^3 + a_2 r^2\right)}{\log(r)}$$

Applying l'Hopital's rule for three times and when $r \to 0$, then

$$d_{cor} = \frac{4 a_2}{2 a_2} = 2.$$

Hence, suppose that $a_2 = a_1 = a_0 = 0$, then $C(r) = a_p r^p + \ldots + a_3 r^3$.

Again, substitute to d_{cor} gives:

$$d_{cor} = \lim_{r \to 0} \frac{\log\left(a_p r^p + a_{p-1} r^{p-1} + \ldots + a_3 r^3\right)}{\log(r)}.$$

Applying l'Hopital's rule for four times, then we get at $r \to 0$:

$$d_{cor} = \frac{18a_3}{6a_3} = 3.$$

As a result, we can conclude if $a_{p-1} = \ldots = a_1 = a_0 = 0.$ and by applying l'Hopital's rule p times on d_{cor} we get $d_{cor} = d = p$ at $r \to 0$.

Bibliography

[1] Al-Kandari, N.M. and Jolliffe, I.T. (2005). Variable selection and interpretation in correlation principal component. *Environmetrics*, **16**, pp 659–672.

[2] Bailer-Jones, C.A.L. (2002). Determination of stellar parameters with GAIA. *Astrophysics and Space Science*, **280**, pp 21–29.

[3] Belkin, M. and Niyogi, P. (2003). Laplacian Eigenmaps for dimensionality reduction and data representation. *Neural Computation*, MIT Press Journals, **15**(6), pp 1373-1396.

[4] Bellman, R.E. (1961). *Adaptive Control Processes: A guided tour*. Princeton University Press.

[5] Bishop, C. (1995). *Neural Networks for Pattern Recognition.* Clarendon Press, Oxford.

[6] Brand, M. (2003). Charting a manifold. In: *Advances in Neural Information Processing Systems*, MIT Press Journals, **15**, pp 961–968.

[7] Brewer, J. and Girolamo, L.D. (2006). Limitation of fractal dimension algorithms with implications for cloud studies. *Atmospheric Research*, **82**, pp 433–454.

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

[9] Camastra, F. and Vinciarelli, A. (2002). Estimating the intrinsic dimension of data with a fractal-based method. *IEEE Transaction on Pattern Analysis and machine intelligence*, **24**(10), pp 1404–1407.

[10] Camastra, F. and Vinciarelli, A. (2001). Intrinsic estimation of data: an approach based on Grassberger-Procaccia's algorithm. *Neural Processing Letters*, **14**(1), pp 27–34.

[11] Carter, K.M., Raich, R. and Hero III, A.O. (2010). On Local Intrinsic Dimension Estimation and Its Applications. *IEEE Transactions on Signal Processing*, **58**(2), pp 650–663.

[12] Chang, C.L. and Lee, R.C.T. (1974). A heuristic relaxation method for nonlinear mapping in cluster analysis. *IEEE Transactions on Computers*,**C-23**, pp 178-184.

[13] Chen, S.S., Keller, J.M. and Crownover, R.M. (1990). Shape from fractal geometry. *Artificial Intelligence*, **43**, pp 199–218.

[14] Chen, P. and Suter, D. (2006). An analysis of linear subspace approaches for computer vision and pattern recognition. *International Journal of Computer Vision*, **68**(1), pp 83-106.

[15] Cumming, J.A. and Wooff, D.A. (2007). Dimension reduction via principal variables. *Computational Statistics & Data Analysis*, **52**, pp 550–565.

[16] Cunningham, P. (2007). Dimension Reduction. *Technical Report UCD-CSI-2007-7*, University College Dublin.

[17] Donoho, D.L. and Grimes, C.E. (2003). Hessian eigenmaps: new locally linear embedding techniques for high-dimensional data. *Proceedings of the National Academy of Arts and Sciences*, **100**, pp 5591–5596.

[18] Dubovikova, M.M., Starchenkoa, N.V. and Dubovikovb, M.S. (2004). Dimension of the minimal cover and fractal analysis of time series. *Physica A*, Elsevier, **339**, pp 591–608.

[19] Duda, R. and Hart, P. (1973). *Pattern Classification and Scene Analysis.* Wiley, New York.

[20] Duong, T. (2004). *Bandwidth selectors for multivariate kernel density estimation.* Thesis, University of Western Australia.

[21] Einbeck, J. (2013). *Lecture Notes "Statistical Methods III".* Academic year 2012/13, Durham University.

[22] Einbeck, J. and Kalantan, Z. (2013). Intrinsic Dimensionality Estimation for High-dimensional Data Sets: New Approaches for the Computation of Correlation Dimension. *Journal of Emerging Technologies in Web Intelligence*, **5**(2), pp 91–97. (doi:10.4304/jetwi.5.2.91–97).

[23] Einbeck, J., Evers, L. and Bailer-Jones, C. (2008). Representing complex data using localized principal components with application to astronomical data. *Principal Manifolds for Data Visualization and Dimension Reduction*, In: *Lecture Notes in Computational Science and Engineering*, **58**, pp 180–204.

[24] Einbeck, J. and Evers, L. (2010). Localized regression on principal manifolds. In: *25th International Workshop on Statistical Modelling (IWSM2010).* 5-9 July 2010, Glasgow, UK.

[25] Einbeck, J. and Evers, L. (2011). LPCM: Local principal curve methods. *R package version 0.44-5.* (see http://CRAN.R-project.org/package=LPCM).

[26] Einbeck, J., Evers, L. and Powell, B. (2010). Data compression and regression through local principal curves and surfaces. *International Journal of Neural Systems*, **20**, pp 177–192.

[27] Einbeck, J., Tutz, G. and Evers, L. (2005). Local principal curves. *Statistics and Computing*, **15**, pp 301–313.

[28] Einbeck, J., Kruger, U. and Kalantan, Z. (2013). On the development of algorithms to estimate the intrinsic dimension of nonlinear non-causal data models. Preprint submitted to *Chemometrics and Intelligent Laboratory Systems Durham University.* Elsevier.

[29] Evers, L. (2013). Local principal manifolds. lpmforge version 0.-0.8, *R package*, University of Glasgow, unpublished version.

[30] Fisher, R. (1936). The use of multiple measurement in taxonomic problems. *Annals of Eugenics*, **7**, pp 179–188.

[31] Frisone, F., Morasso, P., Firenze, F. and Ricciardiello, L. (1995). Application of topology representing networks to the estimation of the intrinsic dimensionality of data. In: *Proceeding of the International Conference on Artificial Neural Networks*, **1**, pp 323–327.

[32] Fukunaga, K. (1971). An Algorithm for Finding Intrinsic Dimensionality of Data. *IEEE Transaction on Computers*, **20**(2), pp 176–183.

[33] Fukunaga, K. (1982). Intrinsic dimensionality extraction. *Handbook of Statistics: Classification, Pattern Recognition and Reduction of Dimensionality*, P.R.Krishnaiah, L.N. Kanal (Eds), **2**, North-Holland, Amsterdam, pp 347–362.

[34] Fukunaga, K. (1990). An introduction to Statistical Pattern Recognition. *Academic Press*. New York.

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

[36] Gorban, A.N., Kégl, B., Wunsch, D.C. and Zinovyev, A.Y. (2008). Preface, Principal Manifolds for Data Visualization and Dimension Reduction. In: *Lecture Notes in Computational Science and Engineering*, **58**, pp VII–VIII.

[37] Härdle, W. (1980). *Smoothing techniques with Implementation in S.* Springer–Verlag.

[38] Hastie, T. (1984). Principal curves and surfaces. *Technical report no.11*, Department of Statistics, Standford University.

[39] Hastie, T. and Stuetzle, W. (1989). Principal curves. *Journal of the American Statistical Association*, **84**(406), pp 502–516.

[40] Hastie, T., Tibshirani, R. and Friedman, J. (2001). *The Elements of Statistical Learning*, Springer, NewYork.

[41] Hein, M. and Audibert, J.Y. (2005). Intrinsic dimensionality estimation of submanifolds in $\mathbb{R}^d$. In: *Proceedings of the 22nd International Conference on Machine Learning*, Bonn, Germany, pp 289–296.

[42] Huber, P. (1985). Projection pursuit. *Annals of Statistics*, **13**(2), pp 435–475.

[43] Höskuldsson, A. (2008). H-methods in applied sciences. *Journal of Chemometrics*, **22**(3-4), pp 150–177.

[44] Hyvärinen, A., Karhunen, J. and Oja, E. (2001). *Independent Component Analysis*. John Wiley & Sons. (http://www.research.ics.aalto.fi/ica/book).

[45] Hyvärinen, A. (2012). Independent Component Analysis: recent advances. *Philosophical Transactions of the Royal Society A*, **371**(1984). (http://www.rsta.royalsocietypublishing.org).

[46] Jackson, D.A. (1993). Stopping rules in principal component analysis: A comparison of heuristical and statistical approaches. *Ecology*, **74**, pp 2204–2214.

[47] Jackson, J.E. (2003). *A Users Guide to Principal Components*. John Wiley & Sons, New York.

[48] Joliffe, I.T. (2002). *Principal Component Analysis*. Springer, New York.

[49] Jutten, C. and Herault, J. (1991). Blind separation of sources, part1: An adaptive algorithm based on neuromimetic architecture. *Signal Processing*, Elsevier B.V., **24**(1), pp 1–10.

[50] Kalantan, Z. and Einbeck, J. (2012). On the computation of the correlation integral for fractal dimension estimation. International Conference on Statistics in Science. Business. and Engineering (ICSSBE2012), *IEEE conference publications*, pp 80–85. (doi 10.1109/ICSSBE.2012.6396531).

[51] Kalantan, Z. and Einbeck, J. (2012). An overview of intrinsic dimension estimation techniques. *Proceedings of the 1st ISM International Statistical Conference 2012*, Johor, Malaysia, pp 516 – 524.

[52] Kégl, B., Krzyżak, A., Linder, T. and Zeger, K. (2000). Learning and Design of Principal Curves. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, **22**(3), pp 281–297.

[53] Kerschen, G. and Golinval, J. (2002). Non–linear generalization of principal component analysis: from a global to a local approach. *Journal of Sound and Vibration*, **254**(5), pp 867–876.

[54] Kevin, M., Raich, R. and Hero III, A.O. (2010). On Local Intrinsic Dimension Estimation and Its Application. *IEEE Transactions on Signal Processing*, **58**(2), pp 650–663.

[55] Kruger, U., Antory, D., Hahn, J., Irwin, G.W. and MnCullough, G. (2005). Introduction of nonlinearity measure for principal component models. *Computers and Chemical Engineering*, **29**(11-12), pp 2355–2362.

[56] Kruger, U. and Xie, L. (2012). *Statistical Monitoring of Complex Multivariate Processes: With Applications in Industrial Process Control.* John Wiley & Sons, Chichester, UK.

[57] Krzanowski, W.J. and Marriott, F.H.C. (1994). *Multivariate Analysis I: Distributions, ordination and inference.* Kendall's library of Statistics, Arnold Publishers, **1**.

[58] Kruger, U., Zhang, J. and Xie, L. (2008). Developments and Applications of Nonlinear Principal Component Analysis-a Review, *Principal Manifolds for Data Visualization and Dimension Reduction*, In: *Lecture Notes in Computational Science and Engineering*, **58**, pp 1–43.

[59] Lee, J.A., Lendasse, A. and Verleysen, M. (2004). Nonlinear projection with curvilinear distances: Isomap versus curvilinear distance analysis. *Neurocomputing.* Elsevier, **57**, pp 49–76.

[60] Levina, E. and Bickel, P. (2005). Maximum likelihood estimation of intrinsic dimension. In: *Advances in Neural Information Processing System*, **17**, pp 777–784.

[61] Liebovitch, L.S. and Toth, T. (1989). A fast algorithm to determine fractal dimensions by box counting. *Physics Letters A*, **141**(8-9), pp 386–390.

[62] Liu, X., Xie, L., Kruger, U., Littler, T. and Wang, S. (2008). Statistical-based monitoring of multivariate non-Gaussian systems. *AIChE Journal*, **54**(9), pp 2379–2391.

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

[64] Malinowski, E.R. (2002). *Factor Analysis in Chemistry.* John Wiley & Sons, New York.

[65] Mandelbrot, B. (1982). *Fractal Geometry of Nature.* Freeman, San Francisco.

[66] Martinetz, T. and Schulten, K. (1994). Topology representing networks. *Neural Networks*, Elsevier Science ltd, **3**, pp 507–522.

[67] Martinez, A.M. and Kak, A.C. (2001). PCA versus LDA. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **23**(2), pp 228–233.

[68] McCabe, G. (1984). Principal Variables. *Technometrics*, **26**(2), pp 137–144. (http://www.jstor.org/stable/1268108)

[69] Mo, D. and Huang, S.H. (2012). Fractal-based intrinsic dimension estimation and its application in dimensionality reduction. *IEEE Transactions on Knowledge and Data Engineering*, **24**(1), pp 59–71.

[70] Originally written for S-Plus by: Kjell Konis and Marco Riani Ported to R by Luca Scrucca (luca@stat.unipg.it) (2012). *forward: Forward search*, R package version 1.0.3. (http://CRAN.R-project.org/package=forward).

[71] Pentland, A. (1984). Fractal -based description of natural scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-**6**(6), pp 661–674.

[72] Pettis, K.W., Baily, T.A. and Jain, A.K. (1979). An intrinsic Dimensionality Estimator from Near-Neighbor Information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-**1**(1), pp 25–37.

[73] Pomeroy, J.W. and Schmidt, R.A. (1993). The use of fractal geometry in modeling intercepted snow accumulation and sublimation. In: *Proceeding 50th Anniversary Meeting of Eastern snow Conference*, pp 1–9.

[74] R Core Team, (2012). R: A language and environment for statistical computing. *R Foundation for Statistical Computing*, Vienna, Austria. ISBN 3-900051-07-0. (URL http://www.R-project.org/).

[75] Raginsky, M. and Lazebnik, S. (2005). Estimation of intrinsic dimensionality using high-rate vector quantization. *Advances in Neural Information Processing Systems*, **19**, pp 352–356.

[76] Roweis, S. and Saul, L. (2000). Nonlinear Dimensionality Reduction by Locally Linear Embedding. *SCIENCE*, **290**, pp 2323–2326. (www.sciencemag.org).

[77] Rummel, D. (2005). The relevance vector machine under covariate measurement error. *Classification—The Ubiquitous Challenge*, In: C. Weihs & W. Gaul (eds), Springer, pp 296-303.

[78] Schleicher, D. (2007). *Hausdorff Dimension, Its properties, and Its Surprises.* The American Mathematical Monthly, **114**(6), pp 509-528.

[79] Scholz, M., Fraunholz, M. and Selbig, J. (2008). Nonlinear Principal Component Analysis: Neural Network Models and Applications. *Principal Manifolds for Data Visualization and Dimension Reduction.* In: *Lecture Notes in Computational Science and Engineering*, **58**, pp 45–68.

[80] Schölkopf, B., Smola, A. and Müller, K. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, MIT Press Journals, **10**, pp 1299-1319.

[81] Scott, D.W. (1992). *Multivariate Density Estimation: Theory, Practice, and Visualization*, Wiley Series in Probability and Mathematical Statistics, John Wiley & Sons, Canada.

[82] Stone, M. (1974). Cross-Validatory Choice and Assessment of Statistical Prediction. *Journal of the Royal Statistical Society (Series B)*, **36**, pp 111–147.

[83] Takens, F. (1985). On the numerical determination of the dimension of an attractor. *Dynamical System and Bifurcations*, *Lecture Notes in Mathematics*, Springer-Verlag, **1125**, pp 99–106.

[84] Tenenbaum, J.B., Silva, V. and Langford, J.C. (2000). A global Geometric Framework for Nonlinear Dimensionality Reduction. *SCIENCE*, **290**, pp 2319–2323.

[85] Theiler, J. (1990). Estimating Fractal Dimension. *Journal of the Optical Society of America*, **7**(6), pp 1055–1073.

[86] Theiler, J., Eubank, S., Longtin, A., Galdrikian, B. and Farmer, J.D. (1992). Testing for nonlinearity in time series: the method for surrogate data. *Physica D*, **58**, pp 77–94.

[87] Valle, S., Li, W. and Qin, S.J. (1999). Selection of the Number of Principal Components: The Variance of the Reconstruction Error Criterion with a Comparison to Other Methods. *Industrial & Engineering Chemistry Research*, **38**(11), pp 4389–4401.

[88] Vaswani, N. and Chellappa, R. (2006). Principal Components Null Space Analysis for Image and Video Classification. *IEEE Transactions on Image Processing*, **15**(7), pp 1816-1830.

[89] Verveer, P.J. and Duin, R. (1995). An evaluation of intrinsic dimensionality estimators. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, **17**(1), pp 81–86.

[90] Wang, P.S (2011). *Pattern Recognition, Machine intelligence and Biometrics.* Springer.

[91] Wong, A., Wu, L., Gibbons, P.B. and Faloutsos, C. (2005). Fast estimation of fractal dimension and correlation integral on stream data. *Information Processing Letters*, **93**, pp 91–97.

[92] http://www.spiedl.org/terms.

[93] http://www.scholarpedia.org/article/Grassberger-Procaccia-algorithm.

[94] Yin, H. (2002). ViSOM-A novel method for multivariate data projection and structure visualisation. *IEEE Transaction on Neural Networks*, **13**(1), pp 237-243.

[95] Yin, H. (2008). Learning Nonlinear Principal Manifolds by Self-Organising Maps. *Principal Manifolds for Data Visualization and Dimension Reduction.* In: *Lecture Notes in Computational Science and Engineering*, **58**, pp 69–96.

[96] Zhang, D., Samal, A. and Brandle, J.R. (2007). A method for estimating fractal dimension of tree crowns from digital images. *CSE Journal Articles.* Paper 97.