Robustness in Dimensionality Reduction

by

Jiaxi Liang

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Statistics

Waterloo, Ontario, Canada, 2016

© Jiaxi Liang 2016

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Dimensionality reduction is widely used in many statistical applications, such as image analysis, microarray analysis, or text mining. This thesis focuses on three problems that relate to the robustness in dimension reduction.

The first topic is the performance analysis in dimension reduction, that is, quantitatively assessing the performance of a algorithm on a given dataset. A criterion for success is established from the geometric point of view to address this issues. A family of goodness measures, called *local rank correlation*, is developed to assess the performance of dimensionality reduction methods. The potential application of the local rank correlation in selecting tuning parameters of dimension reduction algorithms is also explored. The second topic is the sensitivity analysis in dimension reduction. Two types of influence functions are developed as measures of robustness, based on which we develop graphical display strategies for visualizing the robustness of a dimension reduction method, and flagging potential outliers. In the third part of the thesis, a novel robust PCA framework, called *Performance-Weighted Bagging PCA*, is proposed from the perspective of model averaging. It obtains a robust linear subspace by weighted averaging a collection of subspaces produced by subsamples. The robustness against outliers is achieved by a proper weighting scheme, and possible choices of weighting scheme are investigated.

Acknowledgments

Foremost, I would like to express my sincere gratitude to my mentors Dr. Christopher Small and Dr. Shojaeddin Chenouri. With their wisdom, inspiration, and trust, they have guided me through my path of pursuing the PhD, and helped me overcome any obstacles on the way. Their support to me is beyond research, and I will forever cherish their guidance. I also extend thanks to my committee, Dr. Mu Zhu, Dr. Ali Ghodsi, Dr. Edward Vrscay, and Dr. David Tyler, for their valuable suggestions and insightful comments. I have certainly benefited a lot from the discussion with them.

I also want to thank all my good friends. Special thanks go to Marco Shum and Lu Xin for being such great office mates and friends, to Zhiyue Huang for the inspiring discussion and his suggestions, and to Rui Fu for his encouragement. Also thanks to Min Chen, Jay Gweon, Feng He, Celia Huang, Daniel Severn, Xichen She, Hua Shen, Chunlin Wang, Chengguo Weng, Ying Yan, Yujie Zhong for their friendship, and thanks to any other friends who care enough to read this thesis.

I am also very grateful to the support staff of the department. I especially want to thank Mary Lou Dufton and Leanne Bird, for their kind help with all kinds of problems I had.

I would also like to thank Dr. Joel Dubin, Dr. Bin Li, Dr. Paul Marriott, Dr. Greg Rice, Dr. Stefan Steiner, and Dr. Changbao Wu, for both their encouragement, and joyful moments on soccer and basketball fields. I will definitely miss those games.

Last but not the least, I feel extremely grateful to my loving parents, Feibao Liang and Meiying Zheng, for always supporting and understanding me. My love to them is beyond words.

Dedication

This thesis is dedicated to my wonderful parents, Feibao Liang and Meiying Zheng.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

AR	Agreement rate metric
Bagging	Bootstrap aggregating
BP	Breakdown point
EIF	Empirical influence function
IF	Influence function
KPCA	Kernel Principal component analysis
LCMC	Local continuity meta criterion
LLE	Local linear embedding
LRC	Local rank correlation
LTSA	Local tangent space alignment
MDS	Multidimensional scaling
MRREs	Mean relative rank errors

MSD	Mean square distance
MVU	Maximum variance unfolding
PCA	Principal component analysis
PCP	Principal component pursuit
PP	Projection pursuit
RV	Residual variance
SC	Sensitivity curve
SIF	Sample influence function
SVD	Singular value decomposition
T & C	Trustworthiness and continuity

Chapter 1

Introduction and Overview

1.1 Review of robustness

Robust statistics are a class of statistical procedures which are created to deal with an instability problem of the optimal procedures of classical methods. The use of term “robustness” can be traced back to George Box [1953], who discussed in his paper the sensitivity to non-normality of some statistical tests. Subsequently, John Tukey [1960; 1962], Peter Huber [1964; 1965; 1981], and Frank Hampel [1968] gave respective contributions toward the foundations of robust statistics. Since then it has been systematically investigated and developed by many other researchers. Nowadays robust statistics are an important alternative to the classical approaches. To better understand this concept, three questions must be answered: What is robustness? Why use robust procedures? And how does one measure robustness?

1.1.1 What is robustness and why it is needed

All statistical models and methods include assumptions on the underlying distribution, such as normality, linearity, independence, etc. These assumptions are either for mathematical convenience or based on previous experience, and an inference one makes relies not only on the observed data, but also on these assumptions. Most of traditional statistical methods are optimal only when their assumptions are satisfied.

However, it is generally understood that these assumptions are at most an approximation to reality. In practice, the appearance of outlying data caused by either gross error (large errors in measurement, see Dixon [1953]; Grubbs [1969]) or flaws in model assumptions often occur. Thus, besides considering criteria of optimality, it is natural for one to expect that a statistical procedure is stable in the sense that a small departure from the model assumptions or a small proportion of outlying data will only cause a small error in the final conclusion, so that the procedure can still provide a good result.

Unfortunately, statistical procedures are not always stable. In the 19th and 20th centuries, statisticians (such as S. Newcomb, K. Pearson, H. Jeffreys, and E. S. Pearson) were aware of the instability of some traditional methods, and in recent decades, studies point to the fact that even some commonly used statistical procedures are excessively sensitive to seemingly minor deviations from model assumptions. Sometimes these deviations will even cause the breakdown of the procedure (Hampel [2001]).

In many location parameter estimation problems, the sample mean is a common choice. The following example illustrates the performance of the sample mean when model assumptions are violated.

Example 1.1. *Robustness of sample mean*: Suppose we have a sample of observations x_i, $i = 1 \dots n$. We assume they are iid from $N(\mu, \sigma^2)$. In order to estimate the location μ, the traditional choice is the sample mean, which is unbiased and attains the smallest variance at the assumed model. However, if the normality assumption is not exactly correct, the performance of the sample mean might not be good (as measured by the variance and bias).

Let the observations come from the following distribution:

$$X_i \sim \begin{cases} N(\mu, \sigma^2) & \text{with probability } 1-\epsilon \\ f(x) & \text{with probability } \epsilon \end{cases},$$

where $f(x)$ is some non-Normal distribution with mean θ and variance τ^2. Then it is easy to calculate that

$$\begin{aligned} &\text{Bias}(\overline{X}) = \epsilon(\theta - \mu)\,, \\ &\text{Var}(\overline{X}) = (1-\epsilon)\frac{\sigma^2}{n} + \epsilon\frac{\tau^2}{n} + \frac{\epsilon(1-\epsilon)(\theta-\mu)^2}{n}\,. \end{aligned}$$

Even if ϵ is very small (i.e. X_i is approximately Normal), the variance and bias of $\overline{X}$ could become arbitrarily large or even infinite (e.g. when $f(x)$ is the Cauchy pdf).

As illustrated in the above example, the optimal estimator (in the sense of efficiency) for the Normal location parameter, will provide a poor result even under a slightly contaminated model.

This fact motivates statisticians to create the concept of “robustness” to study the effect of wrong assumptions on a given statistical method. Robustness is considered to be a property of a statistical procedure, which describes its behavior under violations of the model assumptions. As stated by Huber [1981], "robustness signifies the insensitivity to small deviations from assumptions". The effect of wrong assumptions is not limited to the parameter estimation problem (Example 1.1), the importance of studying robustness of statistical procedures is recognized by statisticians in hypothesis tests (Gastwirth and Rubin [1971]; He et al. [1990]; He [1991]), regression (Rousseeuw [1984]; Ruppert and Carroll [1971]), mixed models (Welsh and Richardson [1997]), and almost all areas of statistical research. Furthermore, many robust procedures were proposed as reliable alternatives to traditional procedures, aiming to provide reasonable results when the model assumptions are not exactly satisfied.

Alternatively, to avoid the study of robustness, it might be possible to build up a diagnostics system to test the model assumptions and clean the data before applying the traditional methods. Based on this idea, many detection techniques were developed (e.g. outlier rejection, normality tests. See Hawkins [1980]; Hodge and Austin [2004]), but unfortunately these rules are not enough to replace the role of robustness for several reasons.

First of all, traditional outlier detection methods have difficulties in dealing with *masking* and *swamping* (See Fieller [1976]; Barnett and Lewis [1984]). It is discussed by some researchers that even the best rejection rules do not achieve the expectation of completely identifying the violation of model assumptions (Hampel et al. [1986]; Hampel [1974]). Secondly, the criteria for detection and rejection are usually subjective so these rules often suffer from false rejection and false retention error, which will cause a significant loss of efficiency (Hampel [1985]). Thirdly, lots of diagnostics approaches are closely related or even based on robust methods (Barnett and Lewis [1984]; Gather and Becker [1997]). Moreover, when the diagnostics show that some assumptions do not hold (e.g. independence), there does not always exist a well-established alternative approach to deal with the violation.

Thus, robustness provides a safeguard against the situation when the model is not exactly true but this fact is difficult to be detected.

1.1.2 Contamination models and outliers

The uncertainty in the model assumption results in the discordant observation in the sample, usually referred as "outlier" (Beckman and Cook [1983]). An outlier is defined as "one that appears to deviate markedly from other members of the sample in which it occurs" (Grubbs [1969]). There are two main reasons for the appearance of outliers.

- Case I. The presence of gross error. In this case, one believes that the majority of the observations comes from the assumed model, while the remaining observations are from outside the population being examined. These contaminated observations are considered to be bad for the inference because they are non-informative for the assumed model and may corrupt the clean data.

- Case II. The model misspecification. In this case, the assumed model oversimplifies or provides an incorrect description of the data. For example, normality is a key assumption for many traditional methods in regression, analysis of variance and multivariate analysis. However this assumption is invalidated if the error distribution has heavy tails (Newcomb [1886]). Mistakenly assuming the normality might have a serious effect on the conclusion.

These two cases lead to two different types of robustness: firstly the robustness against the gross error, and secondly the robustness against the model misspecification (Maronna et al. [2006]). They share the term *robustness* but represent different philosophies. The first one is based on the faith that the model is close enough to the underlying truth, and focuses on the effect of suspicious points on the result, whereas the second one believes the cleanliness of the data, and tries to check the adequacy of model assumptions.

Thus, although outliers can occur due to both gross errors and model misspecification, we will only use the term “outlier” in the first case (i.e. the gross error case) to avoid confusion.

A commonly used tool to formalize the uncertainty of the model assumptions in the distributional setup is the ϵ-contamination model (introduced by Tukey [1962]; Huber [1964]). Let X_i $(i = 1 \dots n)$ be independent random variables with common distribution $\mathcal{H}$ such that

$$\mathcal{H}(x) = (1 - \epsilon) \cdot \mathcal{F}(x) + \epsilon \cdot \mathcal{G}(x) \tag{1.1}$$

where $\mathcal{F}$ is the assumed distribution in the model, $\mathcal{G}$ is a unknown contaminating distribution (possibly with some restrictions), and $0 \leq \epsilon < 1$ is a real number. The contamination level of model (1.1) is usually measured by ϵ, which measures the difference between assumed model $\mathcal{F}$ and true model $\mathcal{H}$. Example 1.1 is an application of the ϵ-contamination model. In addition to ϵ in equation (1.1), one can also use other distance functions defined on distribution space to measure the difference between the assumed model and the true model (See Víšek [1997]; Huber [1981]).

Instead of considering contamination in the distributional sense, we can also define contamination models in a finite sample setup (Donoho and Huber [1983]). Let $\mathbf{x}_n = (x_1, \dots, x_n)'$ be a column vector representing a fixed sample of size n, where $\mathbf{x}'$ denotes the transpose of the matrix or vector $\mathbf{x}$. There are two ways to model the sample contamination:

(i) ϵ-replacement: we replace an arbitrary subset of size m of the original data $\mathbf{x}_n$ by arbitrary values $\omega_1, \ldots, \omega_m$. Let $\mathbf{x}_{(n,m)}$ denote the contaminated sample, and the contamination level is $\epsilon = m/n$.

(ii) ϵ-corruption: we adjoin m arbitrary additional points $\boldsymbol{\omega}_m = (\omega_1, \ldots, \omega_m)'$ to the original $\mathbf{x}_n$. Let $\mathbf{x}_{n,m} = \mathbf{x}_n \cup \boldsymbol{\omega}_m$ denote the contaminated sample, and the contamination level is $\epsilon = m/(n+m)$.

Note that although outliers can occur due to both gross errors and model misspecification, we will only use the term "outlier" in the first case, i.e. the gross error case.

1.1.3 Measures of robustness

Considering both optimality and robustness, as argued by Huber [1981] and Hampel et al. [1986], an ideal statistical procedure should satisfy three criteria:

- It should have a high (optimal or nearly optimal) efficiency when the model assumptions hold.
- It should be resistant to slight contamination to the model, i.e. the loss of efficiency due to the variation of assumptions should be as low as possible.
- A small violation of model assumptions will not completely spoil the procedure.

To investigate the second and third points in detail, many different types of measures have been proposed to monitor the decrease of performance of a statistical method due to the contamination. We now discuss these criteria.

Influence function

A widely used and well accepted robust measure is the influence function (IF). This concept was introduced by Hampel in his Ph.D. thesis (Hampel [1968]), and further developed in 1974 (Hampel [1975]). In the paper Hampel considers a statistical estimator as a functional T mapping from the space of probability distributions $\mathcal{F}$ to a parameter space of interest,

$$T : \mathcal{F} \to \mathbb{R}\,.$$

The influence function is defined as the Gâteaux derivative of the functional T, at the assumed distribution $\mathscr{F} \in \mathcal{F}$ and a certain point $x \in \mathbb{R}^d$ based on one-step Taylor expansions of T. It is defined by

$$\begin{aligned} \mathrm{IF}(x;\, T,\, \mathscr{F}) &= \lim_{\epsilon \downarrow 0} \frac{T((1-\epsilon)\mathscr{F} + \epsilon\delta_x) - T(\mathscr{F})}{\epsilon} \\ &= \frac{\partial}{\partial \epsilon} T((1-\epsilon)\mathscr{F} + \epsilon\delta_x)|_{\epsilon\downarrow 0}\,, \end{aligned} \tag{1.2}$$

where δ_x is the point-mass at x.

The influence function is a local concept since it measures quantitatively the change of T according to the point-mass contamination at a certain x. A procedure T is more sensitive (than others) to the contamination δ_x if it has a larger (in absolute sense) value of the influence function at x. Thus, for the sake of robustness, the influence function of a procedure is desired to be bounded. For instance, the classical mean functional

$$T(\mathscr{F}) = \int x d\mathscr{F}\,,$$

has an unbounded IF and is therefore regarded as non-robust.

Hampel also proposed a way to find estimators with optimal efficiency given an upper bound on the influence function (Hampel et al. [1986]). General discussion of influence functions can be found in Serfling [1980], Huber [1981] and Hampel [1974]; Hampel et al. [1986].

Several different types of the finite sample influence function have been developed for use in practice. A direct sample version of influence function, usually called the *empirical influence function* (EIF), is defined similar to the influence function, where the probability distribution $\mathcal{F}$ is replaced with the empirical distribution $\widehat{\mathcal{F}}_n$ of the sample. Another finite-sample version is defined in Tukey [1970], called the *sensitivity curve*, or sometimes also called the *empirical influence function* or *sample influence function.* One adds a virtual observation x to the sample $X = \{x_1, \ldots, x_n\}$ and assesses its influence on the estimate T by

$$\mathrm{SC}(x; T, X) = (n+1)\left\{T(x_1, \ldots, x_n, x) - T(x_1, \ldots, x_n)\right\} .$$

Both the empirical influence function and the sensitivity curve are to evaluate the effect on an estimate of perturbing an observation at a finite sample. Mallow [1975] discussed different definitions of the sensitivity curve under different types of contamination when a new observation is added, the i-th observation is replaced, or deleted. The sensitivity curve is essentially equivalent to the empirical influence function, and it converges to the influence function as $n \to \infty$ (Hampel et al. [1986]; Maronna et al. [2006]).

Breakdown point

The influence function is a useful tool to assess the robustness of a statistical procedure. However, as pointed out in Lindsay [1994], considering only local measures might poorly assess the robustness of some types of estimators. Thus, besides the information about the local behavior of a procedure near the assumed model, one may also want to investigate a global measure which indicates how much assumptions may be violated before the statistical procedure becomes invalid.

The breakdown point (or breakdown value) is a global robustness measure, concerning the extreme situation when the procedure is ruined (called “breakdown”). The concept was first proposed by Hodges [1967] (as “the tolerance of extreme value”) in the location estimation problem. Hampel provided the formal definition of asymptotic breakdown point (Hampel [1968, 1971, 2005]). Donoho [1982] and Donoho and Huber [1983] proposed the finite-sample version breakdown point, and had a general discussion on the application of breakdown points. The intuition beneath the breakdown point is to measure the minimum proportion of contamination in the sample (or in model assumption) that can cause the breakdown of the procedure.

In the framework of Huber’s functional analytic approach to robustness, breakdown is related to the boundedness of statistical functionals. Donoho and Huber [1983] mathematically formalized the “breakdown” phenomenon as that for which the functional is carried beyond the bounds of the parameter space (if it is bounded).

Consider a measurable sample space $(\Omega, \mathcal{B}(\Omega))$ where Ω is the sample space and $\mathcal{B}(\Omega)$ is the Borel σ-algebra. Let $\mathcal{F}$ be the family of all nondegenerate probability measures (or distributions) on $(\Omega, \mathcal{B}(\Omega))$, with a metric d on $\mathcal{F}$ such that

$$\sup_{\mathfrak{F},\mathfrak{H}\in\mathcal{F}} d(\mathfrak{F},\mathfrak{H}) = 1 \tag{1.3}$$

Let $T : \mathcal{F}_T \to \Theta$ be a statistical functional, mapping a subspace $\mathcal{F}_T \subseteq \mathcal{F}$ into some metric space (Θ, D), whereas the metric D satisfies

$$\sup_{\theta_1,\theta_2\in\Theta} D(\theta_1,\theta_2) = \infty \tag{1.4}$$

Definition 1.1. *Asymptotic breakdown point*: For a given functional T, and appropriate metrics D and d on Θ and $\mathcal{F}$ respectively that satisfying (1.3) and (1.4), the asymptotic breakdown point of T at a given distribution $\mathcal{F} \in \mathcal{F}_T$ is defined by

$$\epsilon^*(T,\mathcal{F}) = \inf\left\{\epsilon > 0 : \sup_{d(\mathcal{F},\mathcal{H})<\epsilon} D(T(\mathcal{F}),T(\mathcal{H})) = \infty\right\} \tag{1.5}$$

Example 1.2. *Asymptotic breakdown point of mean*: Let T be the expectation functional

$$T_E(\mathcal{F}) = \int_\Omega x d\mathcal{F}(x).$$

Choose d to be the Kolmogorov-metric

$$d(\mathcal{F},\mathcal{H}) = \sup_{x\in\Omega} |\mathcal{F}(x) - \mathcal{H}(x)|,$$

and D to be the Euclidean metric on Θ ($\Theta = \mathbb{R}$). Then it can be verified (Davies and Gather [2007]) that

$$\epsilon^*(T_E,\mathcal{F}) = 0,$$

for any $\mathcal{F} \in \mathcal{F}_T$, where $\mathcal{F}_T = \{\mathcal{F} : T_E(\mathcal{F}) < \infty\}$.

The asymptotic breakdown point is defined on the probability distribution of the assumed model. Similarly, the finite sample version of breakdown point can be defined on the empirical distribution of the sample.

Definition 1.2. *Finite sample breakdown point*: Given an appropriate metric D, and a sample $\mathbf{x}_n = (x_1, \ldots, x_n)$ of size n, the empirical distribution of $\mathbf{x}_n$ is denoted by $\widehat{\mathcal{F}}_n =$

$\frac{1}{n}\sum_{i=1}^{n}\delta_{x_i}$, where δ_{x_i} is the Dirac measure. The finite sample breakdown point of T at the sample $\mathbf{x}_n$ (or at empirical distribution $\widehat{\mathfrak{F}}_n$), is defined by

$$\text{fsbp}(T, \mathbf{x}_n) = \frac{1}{n}\min\left\{m \in \{1, \ldots, n\} : \sup_{\mathbf{x}_{(n,m)}} D(T(\widehat{\mathfrak{F}}_n), T(\widehat{\mathfrak{Q}}_{(n,m)})) = \infty\right\} \tag{1.6}$$

where $\mathbf{x}_{(n,m)}$ is the sample $\mathbf{x}_n$ with m points replaced by arbitrary value, and $\widehat{\mathfrak{Q}}_{(n,m)} \in \mathcal{F}_T$ is the empirical distribution of $\mathbf{x}_{n,m}$.

Example 1.3. *Finite sample breakdown point of median*: Given a sample $\mathbf{x}_n = (x_1, \ldots, x_n)$ of size n (assume n is odd for convenience), and let T be the sample median functional $T_{med}(\mathbf{x}_n) = x_{((n+1)/2)}$, and Θ, D be the same as in example 1.2. Then it can be verified (Davies and Gather [2007]) that

$$\text{fsbp}(T_{med}, \mathbf{x}_n) = \frac{n+1}{2n}$$

In above definitions, the usual choice of metric D is the Euclidean metric, d is Kolmogorov metric or Prohorov-metric (Huber [1981]), and the statistical functional T represents a particular estimator. Davies and Gather discuss the choice of metric D and d (Davies and Gather [2007]), and also emphasize the importance of affine groups structure (Davies and Gather [2005]) in comparing statistical functionals in terms of breakdown points.

Definition 1.1 and 1.2 are well accepted because of their simplicity and intuition. They have been applied in location and scale parameter estimation and linear regression problem (Yohai [1987]; Donoho and Gasko [1992]; Ellis and Morgenthaler [1992]; Davies [1993]; Müller and Uhlig [2001]). The concept of breakdown point is also generalized to cope with other statistical problems including testing (He [1991]; He et al. [1990]), multivariate analysis (Rousseeuw [1985]; Gordaliza [1991]; Lopuhaä [1992]; He and Fung [2000]), directional data (He and Simpson [1992]), nonlinear regression (Stromberg and Ruppert [1992]), and time series (Lucas [1997]; Mendes [2000]; Ma and Genton [2000]; Genton [2003]).

However, the standard definition has some limitations (Genton and Lucas [2003]), and attempts have been made to obtain a more general definition of breakdown points by formalizing the concept of "breakdown" from different perspectives (Sakata and White [1995]; He and Simpson [1993]; Genton and Lucas [2003]).

Among the affine equivariant estimators, it is possible to calculate the upper bound of the breakdown point in many statistical problems (Davies and Gather [2007]), and estimators with highest possible breakdown points are develeped. However, to fully understand the robustness of a statistical procedure requires the combination of different types of robustness measures. Simply pursuing the highest possible breakdown point may be misguided (see Huber and Ronchetti [2011]; He and Portnoy [2000]).

Other robustness measures

The influence function and breakdown point consider the extreme situations of contamination. The influence function considers infinitesimal values of ϵ and the breakdown point seeks for the smallest contamination level $\epsilon^\star$ under which a procedure becomes invalid. Huber [1964] proposed another measure which allows one to study the behavior of a procedure under a fixed contamination level ϵ (before breakdown). He introduced several distance-based neighborhoods to model the deviation from assumptions, and further considered the worst asymptotic performance (in the sense of variance or bias) of a procedure under a certain contamination level. These measures are known as minimax asymptotic variance (MV) and minimax bias (MB), or sometimes referred as maximum asymptotic variance and maximum bias (Huber [1964, 1981]). Using the contamination model (1.1), the MB and MV of the statistical functional T at a distribution $\mathcal{F} \in \mathscr{F}$ are defined as

$$\mathrm{MB}(\mathcal{F}, T, \epsilon) = \sup_{\mathcal{G} \in \mathscr{F}} \left| T\left((1-\epsilon)\mathcal{F} + \epsilon\mathcal{G}\right) - \theta \right| \tag{1.7}$$

$$\mathrm{MV}(\mathcal{F}, T, \epsilon) = \sup_{\mathcal{G} \in \mathscr{G}} \mathrm{Var}\left[T\left((1-\epsilon)\mathcal{F} + \epsilon\mathcal{G}\right)\right] \tag{1.8}$$

Based on this idea, Huber developed a class of robust procedures whose worst asymptotic performance is minimized (called the minimax approach). It has been shown that these approaches have some good finite sample properties (see Andrews et al. [1972]). Other similar measures based on the same idea include the bias curve (Rousseeuw and Croux [1994]), contamination sensitivity and gross-error sensitivity (Hampel [1968]).

Besides these quantitative measures, Hampel [1971] also introduced the concept of qualitative robustness which is closely related to the influence function and the breakdown point. Since Hampel, further theoretical development has been made by Cuevas [1988], and applications can be found in Lambert [1982]; Rieder [1982]; Boente et al. [1987]; Papantoni-Kazakos [1984].

1.2 Introduction to dimensionality reduction

With the development of data collection and storage capabilities, researchers across a wide variety of fields are facing larger and larger datasets with increasing dimensionality, such as images, videos, fingerprints, text documents, etc. Higher dimensionality brings challenges together with benefits. More variables provide more information for inference, but at the same time, the size of data needed for a reliable result increases exponentially with the dimensionality (See Bellman [1961]; Donoho [2000]), therefore traditional statistical methods have difficulties to cope with the explosive growth of dimensionality.

To solve this, dimensionality reduction methods have been developed and applied as pre-processing tools to deal with such high-dimensional datasets. The foundation of dimensionality reduction is the belief that there exists some underlying (unknown) geometric structure in the observed high-dimensional data which allows us to use a lower dimensional representation to characterize the data without losing this structure. Thus, the purpose of the dimensionality reduction methods is to reveal this structure.

Revealing the low-dimensional representation not only improves the efficiency of computation, but also enhances the understanding of the nature of the data. Over the last few decades, many dimensionality reduction algorithms have been proposed. Summaries and surveys can be found in many books and papers (Carreira-Perpinan [1997]; Friedman et al. [2009]; Fodor [2002]) and new ideas are still being contributed to the area. To better illustrate these ideas, we need to first introduce some geometric concepts and notations.

1.2.1 Topology and manifolds

We assume the readers have enough background knowledge about topology, topological space and geometry, and are familiar with basic concepts (such as *open sets*, *neighborhood*, *C^∞ maps*, *connected spaces*, *homeomorphism*, *tangent spaces*, etc. Mathematical details can be found in Kelley [1955]; Armstrong [1979]; Lee [2000, 2003]) .

Manifold

Suppose $\mathcal{M}$ is a topological space. We say $\mathcal{M}$ is a topological d-manifold if it satisfies the following conditions:

- $\mathcal{M}$ is a *Hausdorff space*: For all p, $q \in \mathcal{M}$, there exist disjoint open subsets $U, V \subset \mathcal{M}$ such that $p \in U$ and $q \in V$.

- $\mathcal{M}$ is *second countable*: There exists a countable basis for the topology of $\mathcal{M}$.
- $\mathcal{M}$ is *locally Euclidean of dimension d*: For all $p \in \mathcal{M}$, there exist open sets $U \subset \mathcal{M}$, $\widetilde{U} \subset \mathbb{R}^d$ such that $p \in U$ and there exists a homeomorphism $\phi : U \to \widetilde{U}$.

The dimensionality of $\mathcal{M}$ is d, and the manifold is denoted as $\mathcal{M}^d$ in this paper if we want to emphasize its dimensionality.

A *chart* for a topological space $\mathcal{M}$ is a homeomorphism ϕ from an open subset $U \subset \mathcal{M}$ to an open subset in Euclidean space, it is usually denoted as (U, ϕ). Based on this notion, we can define two important concepts:

- *Atlas*: An atlas $\mathcal{A}$ for a topological space $\mathcal{M}$ is a collection of charts $\mathcal{A} = \{(U_\alpha, \phi_\alpha)\}$ on $\mathcal{M}$ such that $\bigcup U_\alpha = \mathcal{M}$.
- *Transition maps*: Provide two charts (U_α, ϕ_α) and (U_β, ϕ_β) for a topological space $\mathcal{M}$ such that $U_\alpha \cap U_\beta \neq \emptyset$, the transition map $\tau_{\alpha,\beta} : \phi_\alpha(U_\alpha \cap U_\beta) \to \phi_\beta(U_\alpha \cap U_\beta)$ is defined as:

$$\tau_{\alpha,\beta} = \phi_\beta \circ \phi_\alpha^{-1}.$$

A topological manifold $\mathcal{M}$ is said to be a differentiable manifold if it is equipped with an atlas $\mathcal{A} = \{(U_\alpha, \phi_\alpha)\}$ such that for all $\phi_\alpha, \phi_\beta \in \mathcal{A}$, the transition map $\tau_{\alpha,\beta}$ is differentiable.

An atlas satisfies such conditions is called a *differentiable structure* on $\mathcal{M}$. A simple example of differentiable manifold is the Euclidean space $\mathbb{R}^d$.

Embedding and embedded submanifolds

Given a differentiable manifold $\mathcal{N}^D$, a d-submanifold $\mathcal{M}^d \subset \mathcal{N}^D$ $(0 < d \leq D)$ is called an *embedded submanifold* of $\mathcal{N}^D$ if for every point $p \in \mathcal{M}$, there exists a chart

$(U \subset \mathcal{N}, \phi : U \to \mathbb{R}^D)$ such that $p \in U$, ϕ is a diffeomorphism, and $\phi(\mathcal{M} \cap U)$ is a d-flat in $\mathbb{R}^D$.

In dimensionality reduction problems, manifolds can be characterized as embedded submanifolds of Euclidean space by Whitney embedding theorem (Whitney [1936]). A simple example is that a m-sphere $\mathcal{S}^m$ is an embedded submanifold of $\mathbb{R}^{m+1}$. In this paper, we use the term "manifold" to refer the (differentiable) embedded submanifold of some Euclidean space unless specified otherwise, i.e. $\mathcal{M}$ means $\mathcal{M}^d \subset \mathbb{R}^D$ (for some $d \leq D$).

Suppose $\mathcal{M}$ and $\mathcal{N}$ are two differentiable manifolds, and $f : \mathcal{M} \to \mathcal{N}$ is a C^∞ map. The mapping f is said to be an *embedding* if it satisfies:

- f is an immersion: the derivative of f is injective at every point $p \in \mathcal{M}$.
- f is a homeomorphism onto its image $f(\mathcal{M}) \subset \mathcal{N}$ in the subspace topology.

The image $f(\mathcal{M})$ is called an *immersed submanifold* of $\mathcal{N}$, and $\mathcal{M}$ is said to be embedded in $\mathcal{N}$ by the mapping f.

Riemannian manifold and geodesics

Consider a point p on a differentiable manifold $\mathcal{M}$. All vectors that are tangent to $\mathcal{M}$ at the point p will form a vector space $T_p(\mathcal{M})$ called tangent space at p. Suppose that for every point p on $\mathcal{M}$, the tangent space has an inner-product $g_p = \langle \cdot, \cdot \rangle : T_p(\mathcal{M}) \times T_p(\mathcal{M}) \to \mathbb{R}$. The collection of inner-products $g = \{g_p | p \in \mathcal{M}\}$ is called a *Riemannian metric* on $\mathcal{M}$, and the manifold equipped with a Riemannian metric is called a *Riemannian manifold*, denoted as $(\mathcal{M}, g)$.

Given a manifold $\mathcal{M}$, and two points $p, q \in \mathcal{M}$, a smooth curve on $\mathcal{M}$ from p to q is defined as a continuous map $\zeta : I \to \mathcal{M} \subset \mathbb{R}^D$, where $I = [a, b] \subset \mathbb{R}$, $\zeta(a) = p$ and

$\zeta(b) = q$. The smooth curve joins points p and q are not unique, let $\mathcal{C}_{pq}$ be the collection of all such curves. With the metric g defined on $\mathcal{M}$, we can calculate the length of any smooth curve on the manifold:

$$L(\zeta) = \int_a^b \sqrt{g(\zeta'(t), \zeta'(t))}dt,$$

and the distance between p and q on $\mathcal{M}$ is defined as

$$d^{\mathcal{M}}(p, q) = \inf_{\zeta \in \mathcal{C}_{pq}} L(\zeta).$$

The distance $d^{\mathcal{M}}(p, q)$ is called the *geodesic distance* on the manifold $\mathcal{M}$. If $\mathcal{M}$ is a Euclidean space, the geodesic distance would be the Euclidean distance.

1.2.2 Problem setup

In dimensionality reduction, we assume that the observed data in high-dimensional space lie on (or near) an embedded submanifold with lower dimensionality. With this fundamental assumption, it is possible to represent the high-dimensional data in a lower-dimensional space. Formally, we state the problem as following: suppose that there are n data points in a q-dimensional space $\mathbb{R}^q$, denoted by a set of column vectors $\{\mathbf{y}_1, \ldots, \mathbf{y}_n\}$, or together by an $n \times q$ matrix $\mathbf{Y}$ with j-th row being the transpose of the j-th data point $\mathbf{y}'_j$. Further assume $\mathbf{y}_j$ are mapped into a higher-dimensional space $\mathbb{R}^p$ by an unknown smooth embedding $\varphi : \mathbb{R}^q \to \mathcal{M}^q \subset \mathbb{R}^p$ $(p > q)$ possibly with noise:

$$\mathbf{x}_j = \varphi(\mathbf{y}_j) + \epsilon_j, \tag{1.9}$$

where $\epsilon_j \in \mathbb{R}^p, j = 1, \ldots, n$ is the noise with mean 0. We refer "case j" as the index of the correponding points $\mathbf{x}_j$ and $\mathbf{y}_j$ in the input and output spaces.

We only observe $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ in $\mathbb{R}^p$, together denoted by an $n \times p$ matrix $\mathbf{X}$. We say $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ lie on (or near) the manifold $\mathcal{M}$ with *intrinsic dimensionality* q, or we say the intrinsic dimensionality of $\mathbf{X}$ is q. The purpose of dimensionality reduction algorithms are essentially trying to reconstruct the inverse mapping $\psi = \varphi^{-1}$, and to recover $\mathbf{y}$ by $\widehat{\mathbf{y}} = \widehat{\psi}(\mathbf{x})$ (sometimes we only recover $\widehat{\mathbf{y}}$ with implicit $\widehat{\psi}$). We denote a given dimensionality reduction method as a mapping $\psi : \mathbb{R}^p \to \mathbb{R}^q$ in the rest part of the thesis. In many methods, ψ is a function of the entire observed dataset, so we also write the low-dimensional configuration $\mathbf{Y}$ as $\mathbf{Y} = \psi(\mathbf{X})$ for convenience.

This type of dimensionality reduction can be also viewed as learning the structure of embedded submanifold $\mathcal{M}^q$, so it is also called manifold-learning. In this thesis, we shall restrict our attention to manifold-learning. Different algorithms have different assumptions on ψ. According to these assumptions we can roughly classify the dimensionality reduction algorithms into two major types: linear methods and non-linear methods.

1.2.3 Linear methods

In general, linear dimensionality reduction methods assume that the embedded subspace is a linear subspace, and look for a linear projection to recover $\mathbf{y}$:

$$\widehat{\mathbf{y}} = \widehat{\psi}(\mathbf{x}) = \mathbf{W}'\mathbf{x} \text{ (or directly } \widehat{\mathbf{Y}} = \mathbf{X}\,\mathbf{W}) \tag{1.10}$$

where $\mathbf{W}$ is a $p \times q$ projection matrix. Choosing $\mathbf{W}$ due to different criteria determines different algorithms.

Principal component analysis (PCA) is a popular and well-known linear method. It was introduced by K. Pearson [1901], and also known in different fields of application as singular value decomposition (SVD), empirical orthogonal functions (EOF), the Karhunen-Loève transform, and the Hotelling transform.

In essence, PCA seeks a linear subspace formed by a set of orthogonal vectors called "principal components" in such a way that the variability of the data is kept as much as possible in the subspace. Given observed data points $\mathbf{X}$ and assuming zero empirical mean, the orthogonal basis (principal components) $\{\mathbf{w}_j\}$ are a set of $p \times 1$ unit vectors that are obtained by

$$\mathbf{w}_1 = \underset{\|\mathbf{w}\|=1}{\arg\max} \left\{ \|\mathbf{X}\mathbf{w}\| \right\},$$

and for $2 \leq k \leq p$

$$\mathbf{w}_k = \arg\max \left\{ \|\mathbf{X}\mathbf{w}\| \,\middle|\, \|\mathbf{w}\| = 1,\, \mathbf{w} \perp \mathbf{w}_j, \forall 1 \leq j \leq k \right\},$$

where $\|\mathbf{w}\|$ is the Euclidean norm of the vector $\mathbf{w}$. In practice, the principal components $\left\{\mathbf{w}_j\right\}$ are calculated by the eigendecomposition of the empirical covariance matrix of the observed data (assuming that the data are centered on the origin, i.e. $\sum_j^n \mathbf{x}_j = 0$):

$$\mathbf{C} = \frac{1}{n} \sum_{j=1}^{n} \mathbf{x}_j \mathbf{x}_j' = \frac{1}{n} \mathbf{X}' \mathbf{X}.$$

The k-th principal component is the eigenvector of $\mathbf{C}$ corresponding to its k-th largest eigenvalue. If the essential assumption that the data actually lie on or near a q-dimensional linear subspace holds (assuming q is known here), we would find that the first q principal components carry most of the variability and then we can disregard the remaining principal components, and project the data onto a low-dimensional space spanned by the orthogonal basis $\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_q)$.

PCA can be also solved in a dual form. The singular value decomposition (SVD) of $\mathbf{X}$ gives

$$\mathbf{X} = \mathbf{U}\mathbf{\Lambda}\mathbf{W}' \,,$$

where columns of $\mathbf{U}$ are top q eigenvectors of $\mathbf{X}\,\mathbf{X}'$, columns of $\mathbf{W}$ are top q eigenvectors of $\mathbf{X}'\,\mathbf{X}$, and the diagonal matrix $\mathbf{\Lambda}$ contains the square roots of eigenvalues of both $\mathbf{X}'\,\mathbf{X}$ and $\mathbf{X}\,\mathbf{X}'$. Note that there exists a one-to-one correspondence between $\mathbf{U}$ and $\mathbf{W}$. Therefore, obtaining $\mathbf{U}_{n\times q}$ and $\mathbf{\Lambda}_{q\times q}$ from the eigendecomposition of $\mathbf{X}\,\mathbf{X}'$ will also lead to the low-dimensional representation of PCA,

$$\begin{aligned} \mathbf{W} &= \mathbf{X}'\,\mathbf{U}\,\mathbf{\Lambda}^{-1} \\ \widehat{\mathbf{Y}} &= \mathbf{X}\,\mathbf{W} = \mathbf{U}\,\mathbf{\Lambda}. \end{aligned}$$

This dual form is typically helpful when the dimensionality p of the input data $\mathbf{X}$ is very large ($p \gg n$).

Multidimensional scaling (MDS) is another classical linear technique which encompasses a collection of methods (Cox and Cox [1994]). Whereas PCA tries to preserve the variability of the data in low-dimensional space, MDS focuses on the pairwise relations (called distance, proximity or dissimilarity) and attempts to provide a geometrical representation of these relations.

MDS takes a pairwise proximity matrix $\mathbf{D} = [d_{ij}]_{n\times n}$ as input, where d_{ij} is a measure of closeness between objects x_i and x_j, and is trying to construct a configuration by minimizing some loss function. There are many versions of MDS algorithm depending on the choice of d_{ij} and loss function, two major classes are metric and non-metric MDS.

Metric MDS chooses d_{ij} to be a metric of the original space (not necessarily Euclidean space), and tries to reconstruct $\mathbf{Y}$ in $\mathbb{R}^q$ by

$$\widehat{\mathbf{Y}} = \arg\min_{\mathbf{Y}} \sum\sum_{1 \le i < j \le n} \left(d_{ij} - \hat{d}_{ij} \right)^2,$$

where $\widehat{d}_{ij} = \|\mathbf{y}_i - \mathbf{y}_j\|^2$, and $\|\cdot\|$ is the Euclidean norm. Choosing d_{ij} as Euclidean distance in $\mathbb{R}^p$ will obtain the same result as PCA. In general, MDS can be solved by eigendecomposition of the matrix $\mathbf{D}$, defined by the squared pairwise metric.

In contrast, non-metric MDS tries to preserve the ordinal property of the data rather than the proximity, and the loss function called Stress (Cox and Cox [1994]) is applied.

$$\text{Stress} = \sqrt{\frac{\sum\sum_{1 \le i < j \le n} \left(f(d_{ij}) - \hat{d}_{ij} \right)^2}{\sum\sum_{1 \le i < j \le n} \hat{d}_{ij}^2}},$$

where $f(d_{ij})$ is a monotonic transformation. An iterative algorithm was proposed by R. Shepard [1962] and then refined by J. Kruskal [1964] to minimize Stress and obtain the solution of non-metric MDS.

PCA and MDS are both widely used linear algorithms. However their usefulness is limited by the global linearity of the submanifold. Other linear methods such as factor analysis, projection pursuit, independent component analysis, also share this limitation and cannot provide a satisfactory result if the underlying submanifold does not have the global linearity.

1.2.4 Nonlinear methods

Motivated by the inability of linear methods to capture the nonlinear structure, many nonlinear methods have been developed, including ISOMAP (Tenenbaum et al. [2000]), Local Linear Embedding (Roweis and Saul [2000]), Laplacian Eigenmap (Belkin and Niyogi

[2001, 2003]), Local Tangent Space Alignment (Zhang and Zha [2005]), Self-organizing map (Kohonen [1982, 1990]), Kernel PCA (Schölkopf et al. [1998]), Maximum Variance Unfolding (Weinberger and Saul [2006b,a]), Diffusion Maps (Nadler et al. [2005]), and many different versions of nonlinear PCA (Gnandesikan and Wilk [1969]; Hastie and Steutzle [1989]; Kramer [1991]).

Comparing to the linear methods, the nonlinear methods relax the global linearity assumption about the submanifold, and instead adopt two additional assumptions:

- The embedding $\varphi : \mathbb{R}^q \to \mathcal{M}$ is a *local isometry*: for each $\mathbf{z} \in \mathbb{R}^q$, there exists a neighborhood U of $\mathbf{z}$ such that

  $$d^q(\mathbf{y}_1, \mathbf{y}_2) = d^{\mathcal{M}}(\varphi(\mathbf{y}_1), \varphi(\mathbf{y}_2)),\ \mathbf{y}_1, \mathbf{y}_2 \in U$$

  where d^q is the Euclidean distance in $\mathbb{R}^q$, and $d^{\mathcal{M}}$ is the geodesic distance on $\mathcal{M}$.

- The observed data points are dense enough on the manifold: for each $\mathbf{x} \in \{\mathbf{x}_j\}$, there exists a set $N_{\mathbf{x}}$ of neighboring points such that

  $$d^{\mathcal{M}}(\mathbf{x}, \mathbf{x}_i) \approx d^p(\mathbf{x}, \mathbf{x}_i),\ \mathbf{x}_i \in N_{\mathbf{x}},$$

  where d^p is the Euclidean distance in $\mathbb{R}^p$. The set $N_{\mathbf{x}}$ is called the neighborhood of $\mathbf{x}$.

Given above two assumptions, and equation (1.9), we can have $d^q(\mathbf{y}_i, \mathbf{y}_j) \approx d^p(\mathbf{x}_i, \mathbf{x}_j)$ if $\mathbf{x}_i$ and $\mathbf{x}_j$ are neighboring points. Most of the nonlinear methods are considered to be local methods because they focus on the local geometry of the submanifold, and try to recover $\{\mathbf{y}_j\}$ by preserving the neighborhood relationship. These methods usually consist of a three-step algorithm:

- *Step 1*: Identify the neighborhood N_x for each data point. Usual ways of identifying the neighborhoods are

  K-nearest neighbors $N_{\mathbf{x},K}$ (measured by Euclidean distance)

  ϵ-ball: $N_{\mathbf{x},\epsilon} = \{\mathbf{x}_i \in \{\mathbf{x}_j\} \,|d^p(\mathbf{x}, \mathbf{x}_i) \leq \epsilon\}$

  where k and ϵ are tuning parameters (usually called neighborhood size). Note that in general, $\mathbf{x}_i \in N_{\mathbf{x}_j}$ does not necessarily imply $\mathbf{x}_j \in N_{\mathbf{x}_i}$.

- *Step 2*: Characterize the neighborhood relationship (the relationship is formalized differently in different methods).

- *Step 3*: Construct the low-dimensional configuration that optimally preserve the specified neighborhood relationship.

A typical local method is Local Linear Embedding (LLE). After assigning the neighborhood to each point, LLE characterizes the neighborhood relationship by a set of linear coefficients that reconstruct each data point from its neighbors. The linear coefficients $\{w_{ij}\}$ are obtained by minimizing the reconstruction error:

$$\widehat{W} = \arg\min_{W} \sum_{i}^{n} \left(\mathbf{x}_i - \sum_{j}^{n} w_{ij} \mathbf{x}_j \right)^2 ,$$

where w_{ij} satisfies $w_{ij} = 0$ if $\mathbf{x}_j \notin N_{\mathbf{x}_i}$, and $\sum_j^n w_{ij} = 1$.

Assuming that the coefficients $\widehat{W}$ are invariant to the mapping φ, the same weights $\{\widehat{w}_{ij}\}$ that reconstruct the data point $\mathbf{x}_i$ should also be able to reconstruct the corresponding point $\mathbf{y}_i$ in the embedding space. Thus, the low-dimensional representation $\widehat{\mathbf{Y}}$ is constructed by minimizing the embedding cost function:

$$\widehat{\mathbf{Y}} = \arg\min_{\mathbf{Y}} \sum_{i}^{n} \left(\mathbf{y}_i - \sum_{j}^{n} \widehat{w}_{ij} \mathbf{y}_j \right)^2 ,$$

The properties and limitations of LLE and the complexity of the algorithm are discussed in Roweis and Saul [2000].

Besides the local methods, there is another class of nonlinear methods which consider the global geometry of the submanifold (usually referred as global methods). A typical one in this class is ISOMAP.

ISOMAP considers the global geometry by assuming the mapping φ is an *isometry*, i.e.

$$d^q(\mathbf{y}_1, \mathbf{y}_2) = d^{\mathcal{M}}(\varphi(\mathbf{y}_1), \varphi(\mathbf{y}_2)),\ \mathbf{y}_1, \mathbf{y}_2 \in \mathbb{R}^q,$$

ISOMAP can be viewed as a generalization of metric MDS because it carries the idea of MDS, and tries to preserve the pairwise geodesic distances (instead of Euclidean distances).

The algorithm also starts with identifying the neighborhood for each point. This results in a weighted neighborhood graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the set of vertices $\mathcal{V} = \{\mathbf{x}_j\}$ are the observed data, and the set of edges $\mathcal{E} = \{e_{ij}\}$ indicate the connection between two points. If $\mathbf{x}_i$ and $\mathbf{x}_j$ is assigned as neighboring points (i.e. at least one of them is in the neighborhood of the other), the edge e_{ij} has a weight $w_{ij} = d^p(\mathbf{x}_i, \mathbf{x}_j)$.

The next step is to approximate the pairwise geodesic distance based on the graph $\mathcal{G}$. A *path* P in the graph is defined as a sequence of vertices $P = (v_1, \ldots, v_m)$ such that for all $1 \leq i \leq m-1$, v_i and v_{i+1} are neighboring points. The length of the path P is defined as

$$d^{\mathcal{G}}(P) = \sum_{i=1}^{m-1} w_{i,i+1}.$$

Non-neighboring vertices $\mathbf{x}_v$ and $\mathbf{x}_u$ are connected by any path $P_{uv} = (v_1, \ldots, v_m)$ such that $v_1 = \mathbf{x}_v$ and $v_m = \mathbf{x}_u$. The graph distance (also called shortest path distance) between $\mathbf{x}_v$ and $\mathbf{x}_u$ is defined as

$$d^{\mathcal{G}}_{uv} = \min_{P_{uv}} d^{\mathcal{G}}(P_{uv}).$$

The graph distance can be efficiently calculated by Floyd's algorithm (Floyd [1962]) or Dijkstra's algorithm (Dijkstra [1959]). Then the geodesic distances are approximated by:

$$\widehat{d}_{ij}^{\mathcal{M}} \approx \begin{cases} d^p(\mathbf{x}_i, \mathbf{x}_j) & \text{if } \mathbf{x}_i, \mathbf{x}_j \text{ are neighbors,} \\ d_{ij}^{\mathcal{G}} & \text{if } \mathbf{x}_i, \mathbf{x}_j \text{ are not neighbors.} \end{cases}$$

The final step is to apply metric MDS to reconstruct the low-dimensional configuration $\widehat{\mathbf{Y}}$, with the input being the pairwise geodesic distance matrix $D^{\mathcal{G}} = \left[\widehat{d}_{ij}^{\mathcal{M}}\right]_{n\times n}$.

It has been shown that with some regularity conditions, the graph distance converges to the true geodesic distance as the sample size $n \to \infty$ (Bernstein et al. [2002]). It worth mentioning that ISOMAP additionally assumes *convexity* on the submanifold $\mathcal{M}$. This assumption is important for approximating the geodesic distances, but it appears to be too restrictive in many instances (Donoho and Grimes [2003]). The performance of ISOMAP is discussed in Donoho and Grimes [2002b], and several modified versions have been developed in order to relax the model assumptions (Donoho and Grimes [2002a]; Silva and Tenenbaum [2002]).

1.2.5 Kernel PCA and a unified framework

Kernel PCA (Schölkopf et al. [1997]) is another type of nonlinear dimensionality reduction method. The Kernel PCA performs principal component analysis in a feature space which is related to the original input space by some implicit nonlinear mapping. It is hoped that the structure of the observed data can be unfolded as linear in this high-dimensional feature space.

Assume that there exists a map $\Phi : \mathbb{R}^p \to \mathcal{H}$, transforming the observed data into a

Hilbert space. Define an $n \times n$ matrix $\mathbf{K}$ by

$$\mathbf{K}_{ij} = \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle ,$$

where $\langle \cdot, \cdot \rangle$ is the inner product in the space $\mathcal{H}$. The matrix $\mathbf{K}$ is positive semidefinite, and it is called a *kernel matrix*.

The traditional PCA is then applied on the transformed data $\{\Phi(\mathbf{x}_1), \dots, \Phi(\mathbf{x}_n)\}$. To this end, we consider the eigendecomposition of the covariance operator:

$$\mathbf{C}_\Phi = \frac{1}{n} \sum_{j=1}^{n} \Phi(\mathbf{x}_j) \Phi(\mathbf{x}_j)', \tag{1.11}$$

assuming that $\sum_{j=1}^{n} \Phi(\mathbf{x}_j) = 0$. The low-dimensional subspace is the space spanned by the top q eigenvectors of $\mathbf{C}_\Phi$. The eigenvalue λ and the corresponding eigenvector $\mathbf{v}$ of $\mathbf{C}_\Phi$ are the solutions to the equation

$$\mathbf{C}_\Phi \mathbf{v} = \lambda \mathbf{v}. \tag{1.12}$$

Note that all eigenvectors $\mathbf{v}$ satisfying equation (1.12) and corresponding to eigenvalues $\lambda > 0$ lie in the span of $\{\Phi(\mathbf{x}_1), \dots, \Phi(\mathbf{x}_n)\}$. Thus, rewrite $\mathbf{v}$ as

$$\mathbf{v} = \sum_{i=1}^{n} \alpha_i \, \Phi(\mathbf{x}_i), \tag{1.13}$$

and the problem becomes finding the λ and $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_n)'$.

Substituting equations (1.11) and (1.13) into (1.12), we observe that λ and $\boldsymbol{\alpha}$ satisfy

$$n \, \lambda \boldsymbol{\alpha} = \mathbf{K} \boldsymbol{\alpha}. \tag{1.14}$$

Then, the problem is equivalent to the eigendecomposition of the kernel $\mathbf{K}$. The so-called “kernel trick” allows one to obtain the low-dimensional representation $\widehat{\mathbf{Y}}_{n \times q}$ without specifying the nonlinear map Φ:

$$\widehat{\mathbf{Y}} = \mathbf{A} \boldsymbol{\Lambda}^{\frac{1}{2}},$$

where $\mathbf{\Lambda}$ is a diagonal matrix of the top q eigenvalues of $\mathbf{K}$, and $\mathbf{A} = [\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_q]$ is an $n \times q$ matrix with $\boldsymbol{\alpha}_j$ being the eigenvetor of $\mathbf{K}$ corresponding to the j-th largest eigenvalue.

Also note that, in equation (1.12) we implicitly assume that the transformed data $\{\Phi(\mathbf{x}_1), \ldots, \Phi(\mathbf{x}_n)\}$ have a zero mean. Thus, to validate the above derivation, the transformed data should be centered. This can be guaranteed by an additional centering step on the chosen kernel matrix $\mathbf{K}$, i.e. instead of the chosen kernel $\mathbf{K}$, the eigendecomposition is performed on

$$\widetilde{\mathbf{K}} = (\mathbf{I} - \mathbf{e}\mathbf{e}')\mathbf{K}(\mathbf{I} - \mathbf{e}\mathbf{e}'),$$

where $\mathbf{e} = n^{-1/2}(1, \ldots, 1)'$ is the uniform vector of unit length.

Different choices of the kernel $\mathbf{K}$ will result in different low-dimensional representations. It has been shown that many dimensionality reduction methods, such as MDS, ISOMAP, LLE, Laplacian Eigenmap, and Diffusion maps, can all be described as special cases under the framework of Kernel PCA (Ham et al. [2004]).

For example, ISOMAP is equivalent to Kernel PCA by choosing the kernel

$$\widetilde{\mathbf{K}} = -\frac{1}{2}(\mathbf{I} - \mathbf{e}\mathbf{e}')\mathbf{D}^{\mathcal{G}}(\mathbf{I} - \mathbf{e}\mathbf{e}'),$$

where $\mathbf{D}^{\mathcal{G}}$ is the matrix of squared pairwise geodesic distances. LLE is equivalent to Kernel PCA by choosing the kernel

$$\mathbf{K} = \lambda_{max}\mathbf{I} - (\mathbf{I} - \widehat{\mathbf{W}})'(\mathbf{I} - \widehat{\mathbf{W}}),$$

$$\widetilde{\mathbf{K}} = (\mathbf{I} - \mathbf{e}\mathbf{e}')\mathbf{K}(\mathbf{I} - \mathbf{e}\mathbf{e}'),$$

where $\widehat{\mathbf{W}}$ is the coefficients matrix in LLE algorithm, and λ_{max} is the largest eigenvalue of $(\mathbf{I} - \widehat{\mathbf{W}})'(\mathbf{I} - \widehat{\mathbf{W}})$.

Kernel PCA provides the unified framework of dimensionality reduction, it also provides an interesting insight. The kernel matrix $\mathbf{K}$ is essentially generalized dissimilarity measures between each pair of data points. If one believes that the hidden geometric structure of the observed data can be characterized by the kernel, then different algorithms are simply estimating this kernel in different ways.

1.3 Outline and contributions of the thesis

This thesis covers three topics concerning the robustness in dimension reduction.

In Chapter 2, we tackle the problem of how can we assess the success of a dimension reduction method. The challenge comes from the fact that dimension reduction is stated as an unsupervised problem. A local rank correlation measure is proposed to quantify the performance of dimension reduction methods. The criterion for success in dimension reduction is considered to be the preservation of local isometry in low-dimensional representations. The local rank correlation is easily interpretable, and robust against the presence of outliers. An adjustment is available so that the proposed measure is applicable on the family of output-normalized methods. It is demonstrated in some benchmark datasets that the local rank correlation correctly reflects the performance of a given method. The material in this chapter appears in our submitted paper Liang et al. [2015].

Robustness of any method can be considered against outliers, manifold misspecification, or noise in the data. In Chapter 3 and onwards, we shall focus on robustness against outliers. Specifically, in Chapter 3, the sensitivity analysis in dimension reduction is studied. Two types of influence measures are introduced as tools for studying the robustness of dimension reduction methods. We first define traditional PCA as a functional mapping from the space of p-dimensional distributions to a q-dimensional linear subspace. An empirical influence function of PCA is introduced as the Gâteaux derivative based on a subspace distance measure. This result is generalized to Kernel PCA framework to cope with nonlinear dimension reduction methods. Then, a sample influence function is defined as a supplement based on the local rank correlation from Chapter 2. Chapter 3 also discusses the graphical display strategies for visualizing the influence of a certain point on a given method, and the potential application of influence measures in detecting influential observations.

In Chapter 4, we propose a novel approach, called Performance-Weighted Bagging PCA, to robustify traditional PCA from the perspective of model averaging. Unlike other robust PCA methods which obtain the result from some modified loss functions, the proposed Performance-Weighted Bagging PCA performs traditional PCA on a set of subsamples, and uses the weighted average over subspaces produced by these subsamples. The weighting scheme is the key to make the procedure robust. The local rank correlation from Chapter 2 is a natural but not only candidate. The choice of weighting function is very flexible, and can potentially connect to other robust PCA methods. It is computationally convenient, and robust against outliers. In both simulation studies and surveillance video data, Performance-Weighted Bagging PCA yields competitive results compared to some traditional robust PCA methods.

Chapter 2

Performance Analysis for Dimensionality Reduction

2.1 Introduction

2.1.1 Review of previous work

How to assess and compare the performances of different dimension reduction methods is a challenging issue, and this issue is not yet well explored in the literature. In the supervised learning problems, such as regression or classification, a natural criterion to measure the performance of a given method is defined as the difference between true values and estimated values of response variable, for example prediction error or classification error. However dimensionality reduction, as we state here, is a unsupervised learning problem, which cannot directly use such a criterion to quantify the performance of different dimension reduction algorithms. In order to complete the task, a difference type of goodness measure is needed. This measure is expected to be

- easily interpretable,
- applicable to most algorithms and datasets,
- robust against the presence of outliers,
- robust against misspecification of tuning parameter of the measure.

Many dimension reduction algorithms obtain their results by optimizing given objective functions. One way to assess the performance of a method is to check the value of the corresponding objective function for the output $\widehat{\mathbf{Y}}$. It is only fair, however, to compare different values of tuning parameters of one method, but not appropriate to compare the performance of different methods.

A second possibility is via the residual variance. In PCA, MDS and ISOMAP, the residual variance is usually used in determining the intrinsic dimensionality. It is defined as

$$\mathrm{RV}(\mathbf{X}, \mathbf{Y}) = 1 - r^2(\mathbf{D}_X, \mathbf{D}_Y),$$

where $\mathbf{D}_X$ and $\mathbf{D}_Y$ are the matrices of pairwise distances in $\mathbf{X}$ and $\mathbf{Y}$, respectively, and r is the standard linear correlation coefficient, taken over all entries of $\mathbf{D}_X$ and $\mathbf{D}_Y$. The lower the residual variance, the better input data $\mathbf{X}$ are represented in the embedded space. However, the major concern about this measure is that in nonlinear dimensionality reduction, $\mathbf{D}_X$ is difficult to determine, potentially resulting in an unfair comparison. For example, if $\mathbf{D}_X$ is obtained by the graph distance in ISOMAP, it would automatically imply that ISOMAP is the best algorithm for all datasets.

Another choice is to use the reconstruction error. Recall the problem setup in equation (1.9). For a given method $\psi : \mathbb{R}^p \to \mathbb{R}^q$, the reconstruction error can be written as

$$\mathrm{Err}_{rec} = \mathrm{E}\left\{ \sum_{j=1}^{n} \left(\mathbf{x}_j - \psi^{-1}(\psi(\mathbf{x}_j)) \right)^2 \right\}.$$

This requires the explicit form of the map ψ and its inverse, which is not available for many nonlinear methods.

Recent research focuses on assessing dimensionality reduction methods from the geometric point of view. In the problem setup in Section 1.2.2, we assumed that φ is a local isometry. This assumption implies that the neighboring points in the input space should be mapped to neighbors in the output space, and vice versa for the inverse mapping ψ. This phenomenon can be called "topology preservation". In order to quantify topology preservation, the following notation will be useful.

Notation

For an observed high-dimensional dataset $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \subset \mathcal{M}$ and a low-dimensional representation $\{\widehat{\mathbf{y}}_1, \ldots, \widehat{\mathbf{y}}_n\}$, we have the following notation:

- $\|\cdot\|$: the Euclidean norm.

  $d^{\mathcal{M}}(\cdot, \cdot)$: the geodesic distance on the Riemannian manifold $\mathcal{M}$.

- $|A|$: the cardinality of the set A.

- s_{ij}: the rank of $\|\mathbf{x}_i - \mathbf{x}_j\|$ in ascending order, i.e.

  $$s_{ij} = \left| \{ k : \|\mathbf{x}_i - \mathbf{x}_k\| \leq \|\mathbf{x}_i - \mathbf{x}_j\|, \, 1 \leq k \leq n \} \right| .$$

  r_{ij}: the rank of $\|\mathbf{y}_i - \mathbf{y}_j\|$ in ascending order.

  $\widehat{r}_{ij}$: the rank of $\|\widehat{\mathbf{y}}_i - \widehat{\mathbf{y}}_j\|$ in ascending order.

- $N_{_J}^I(i)$: the index set of J-nearest neighbors of $\mathbf{x}_i$, that is $N_{_J}^I(i) = \Big\{j \,|\, 1 \leq s_{_{ij}} \leq J\Big\}$.

  $N_{_J}^O(i)$: the index set of J-nearest neighbors of $\mathbf{y}_i$, that is $N_{_J}^O(i) = \Big\{j|1 \leq \widehat{r}_{_{ij}} \leq J\Big\}$.

- $\mathcal{N}_{_J}(i) = N_{_J}^I(i) \bigcap N_{_J}^O(i)$.

$\mathcal{N}_{_J}^*(i) = N_{_J}^I(i) \bigcup N_{_J}^O(i)$.

Early attempts to quantify the topology preservation of a dimension reduction method were made in the study of Self-Organizing Maps [Kohonen, 1982]. In order to measure the performance of Self-Organizing Maps, measures such as the *topographic product* [Bauer and Pawelzik, 1992], *topographic function* [Villmann et al., 1997] and *quantization error* [Kaski and Lagus, 1996] have been developed. Advantages and disadvantages of each method are discussed in detail by Pölzlbauer [Pölzlbauer, 2004].

More recently, a few rank-based measures have been proposed, with broader applicability. These include *mean relative rank errors* [Lee and Verleysen, 2007], *trustworthiness and continuity* [Venna and Kaski, 2001], *local continuity meta criterion* [Chen and Buja, 2009], and the *agreement rate metric* [France and Carroll, 2007].

Trustworthiness and continuity (T&C) measures [Venna and Kaski, 2001] are defined by

$$T_{_J} = 1 - \frac{1}{G_{_J}} \sum_{i=1}^{n} \sum_{j \in N_{_J}^O(i) \backslash N_{_J}^I(i)} (s_{_{ij}} - J),$$

$$C_{_J} = 1 - \frac{1}{G_{_J}} \sum_{i=1}^{n} \sum_{j \in N_{_J}^I(i) \backslash N_{_J}^O(i)} (\widehat{r}_{_{ij}} - J),$$

where

$$G_{_J} = \begin{cases} \frac{nJ(2n-3J-1)}{2}, & \text{if } J < n/2 \\ \frac{n(n-J)(n-J-1)}{2}, & \text{if } J \geq n/2 \end{cases}$$

is the normalizing factor.

The *mean relative rank errors* (MRREs) [Lee and Verleysen, 2007] are defined by

$$
\begin{aligned}
M_{J}^{O} &= 1 - \frac{1}{H_{J}} \sum_{i=1}^{n} \sum_{j \in N_{J}^{O}(i)} \frac{\left| s_{ij} - \widehat{r}_{ij} \right|}{s_{ij}}, \\
M_{J}^{I} &= 1 - \frac{1}{H_{J}} \sum_{i=1}^{n} \sum_{j \in N_{J}^{I}(i)} \frac{\left| s_{ij} - \widehat{r}_{ij} \right|}{\widehat{r}_{ij}},
\end{aligned}
$$

where $H_{J} = n \sum_{m=1}^{J} \frac{|n-2m+1|}{m}$ is the normalizing factor.

Both T&C and MRREs are restricted to the interval [0, 1]. Furthermore, higher values of these measures are desirable properties of algorithms. These two measures try to quantify distinguishably two types of topological errors that occur during the dimension reduction procedures,

(i) non-neighboring points in $\mathbb{R}^p$ are mapped by $\widehat{\psi}$ to be neighboring points in $\mathbb{R}^q$,

(ii) neighboring points in $\mathbb{R}^p$ are mapped by $\widehat{\psi}$ to be non-neighboring points in $\mathbb{R}^q$.

These two types of errors create a discrepancy between nearest neighbor ranks in the input and output spaces. Therefore they can be measured by calculating the change of nearest neighbor ranks.

The *agreement rate metric* (AR) [France and Carroll, 2007] and *local continuity meta criterion* (LCMC) [Chen and Buja, 2009] are defined similarly:

$$
\begin{aligned}
\mathrm{AR}_{J} &= \frac{1}{n} \sum_{i=1}^{n} \frac{\left| N_{J}^{O}(i) \cap N_{J}^{I}(i) \right|}{J}, \\
\mathrm{LCMC}_{J} &= \frac{1}{n} \sum_{i=1}^{n} \left(\frac{\left| N_{J}^{O}(i) \cap N_{J}^{I}(i) \right|}{J} - \frac{J}{n-1} \right).
\end{aligned}
$$

As can be seen, AR_J is the average size of the overlap of J-nearest neighborhoods between the low-dimensional reconstruction and the original data. The adjustment in LCMC_J accounts for the expected random overlap. Lee and Verleysen [2009] proposed a co-ranking framework, which includes MRREs, T& C, and LCMC as special cases. Lueks et al. [2011] discussed the co-ranking framework in detail, and provided an extension of the co-ranking framework by introducing an extra tuning parameter to weight rank errors.

Besides the topology preservation measures, Goldberg and Ritov [2009] proposed a Procrustes measure that evaluates how well each local neighborhood matches its corresponding embedding under an optimal linear transformation. It is defined as

$$R(\mathbf{X}, \mathbf{Y}) = \frac{1}{n} \sum_{i=1}^{n} \sum_{j \in N_J^I(i)} \|\mathbf{x}_i - \mathbf{A}_i \mathbf{y}_i - \mathbf{b}_i\|^2 \ .$$

The rotation matrix $\mathbf{A}_i$ is a columns-orthogonal $p \times q$ matrix and the translation vector $\mathbf{b}_i$ is $p \times 1$ vector. They are obtained by solving

$$\underset{\mathbf{A}'\mathbf{A}=\mathbf{I},\, \mathbf{b}\in\mathbb{R}^q}{\arg\min} \left\{ \sum_{j \in N_J^I(i)} \|\mathbf{x}_i - \mathbf{A}\mathbf{y}_i - \mathbf{b}\|^2 \right\} ,$$

where $\mathbf{I}$ is the $p \times p$ identity matrix.

In this definition, $R(\mathbf{X}, \mathbf{Y})$ measures the average local reconstruction error in each neighborhood. A lower value of $R(\mathbf{X}, \mathbf{Y})$ indicates a better low-dimensional representation $\widehat{\mathbf{Y}}$. However, as pointed out in Goldberg and Ritov [2009], a main drawback of $R(\mathbf{X}, \mathbf{Y})$ is that it focuses only on the preservation of distances between neighboring points in $\mathbf{X}$. If more distant points in $\mathbf{X}$ are mapped as neighbors in $\mathbf{Y}$, a relatively low value of $R(\mathbf{X}, \mathbf{Y})$ is obtained as long as distances between neighboring points in $\mathbf{X}$ are generally preserved.

2.1.2 Room for improvement

Although existing topology preservation measures perform reasonably well in many cases, there are certain circumstances which suggest that there is still room for new goodness measures.

The strength of T&C and MRREs is in their ability to distinguish two sorts of undesired errors discussed above. However, as we shall see in Section 2.4, it is hard to calibrate the values of these measures with our visual intuition. There exist examples with similar values for these measures for which some are visually more successful than the others. In addition, as stated in Lueks et al. [2011], in the co-ranking matrix framework, the interpretation of the tuning parameter J is unintuitive, and the evaluation depends dramatically on the choice of J. In the extension of the co-ranking framework (Lueks et al. [2011]), an additional tuning parameter is introduced to achieve a clearer interpretation of tuning parameters. However, the example in the paper shows that the quality assessment still depends heavily on the choice of two tuning parameters, and how to choose values for parameters is not discussed.

Compared to T&C and MRREs, the values of AR and LCMC have a more intuitive interpretation. However, as stated in Lee and Verleysen [2009], AR and LCMC provide less information about the topology preservation in each neighborhood, as illustrated in Figure 2.1. In this plot, 6-nearest neighbors of $\mathbf{x}_1$ are marked with squares, and 6-nearest neighbors of $\mathbf{x}_2$ are marked with diamonds. If we change the neighborhood of $\mathbf{x}_1$ by interchanging positions of $\mathbf{x}_3$ and $\mathbf{x}_4$, the values of AR and LCMC in the neighborhood of $\mathbf{x}_1$ remain the same. This change can only be captured in the neighborhood of $\mathbf{x}_2$.

In addition, a common disadvantage shared by all aforementioned measures arises when we evaluate the performance of output normalized methods, such as Local Linear Embed-

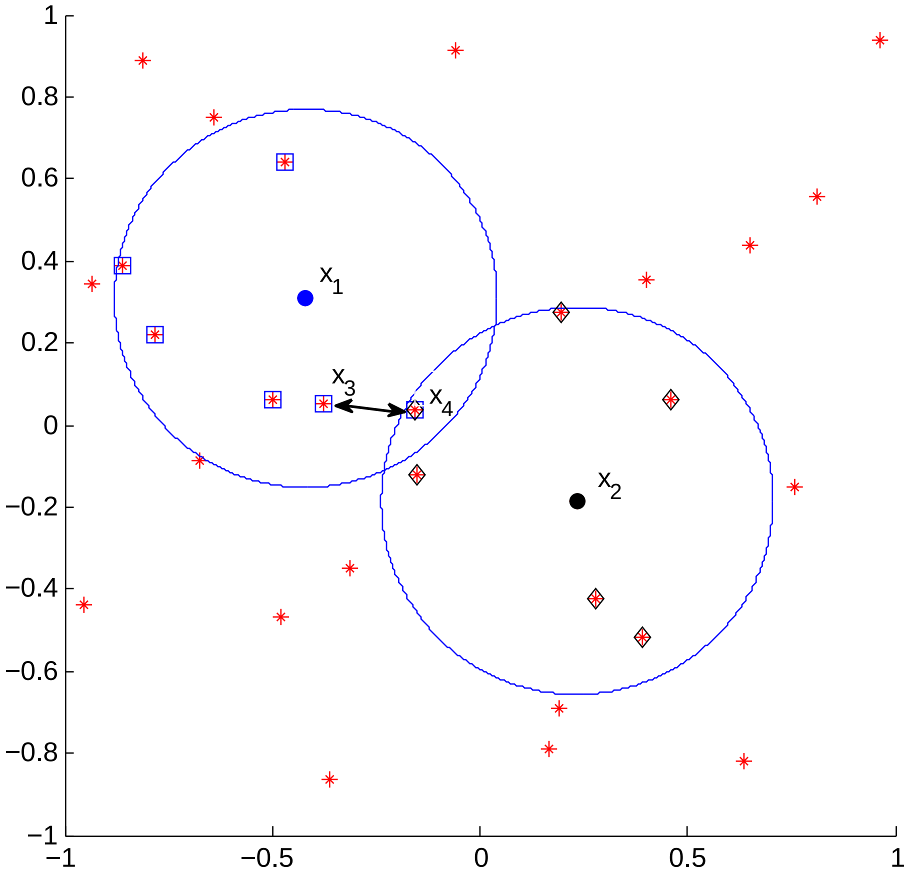

Figure 2.1: Illustration of AR and LCMC.

ding (Roweis and Saul [2000]), Laplacian Eigenmap (Belkin and Niyogi [2001]), and Local Tangent Space Alignment (Zhang and Zha [2005]). Normalization of the output distorts the structure of neighborhoods, and the topological structure will not typically be preserved by the output configuration (Sha and Saul [2005]). As shown in Figure 2.2, the top panels show the configurations before and after normalization. The bottom panels show the neighborhood of the i-th case in two configurations, respectively. Neighboring cases of $\mathbf{x}_i$ are marked with squares, and neighboring cases of $\mathbf{y}_i$ are marked with diamonds.

As can be seen, although two configurations seem to be very similar apart from the fact that the normalization shortens the horizontal distances, it leads to the change of nearest neighbors. Therefore the topology preservation measures typically provide low values for an output normalized method. The Procrustes measure also has difficulties in evaluating output-normalized-methods for the same reason (Goldberg and Ritov [2009]). Goldberg et al. [2008] have pointed out that the low-dimensional configuration $\widehat{\mathbf{Y}}$ from an output-normalized-method can only preserve the topological structure of the input data $\mathbf{X}$ up to an affine transformation. Therefore, if we want to use topology preservation measures to evaluate the performance of output-normalized-methods, an extra adjustment is needed to recover such an affine transformation.

2.2 Naive measures

We first introduce two naive measures, and illustrate why these two measures are not good ways to quantify the performance of dimension reduction methods.

In addition to assumptions in the problem setup (1.9), we assume that for any point $\mathbf{x}_i$, there exists a set $N(i)$ of neighboring cases such that the nearest neighbor ranks of the latent low-dimensional data $\mathbf{Y}$ are preserved in $\mathbf{X}$, i.e.

$$r_{_{ij}} = s_{_{ij}}.$$

Therefore, a low-dimensional representation $\widehat{\mathbf{Y}}$ can be said to have rank fidelity if $\widehat{\mathbf{Y}}$ also preserves such ranks, i.e.

$$\widehat{r}_{_{ij}} = s_{_{ij}}.$$

Two types of errors could occur due to the mapping $\widehat{\psi}$.

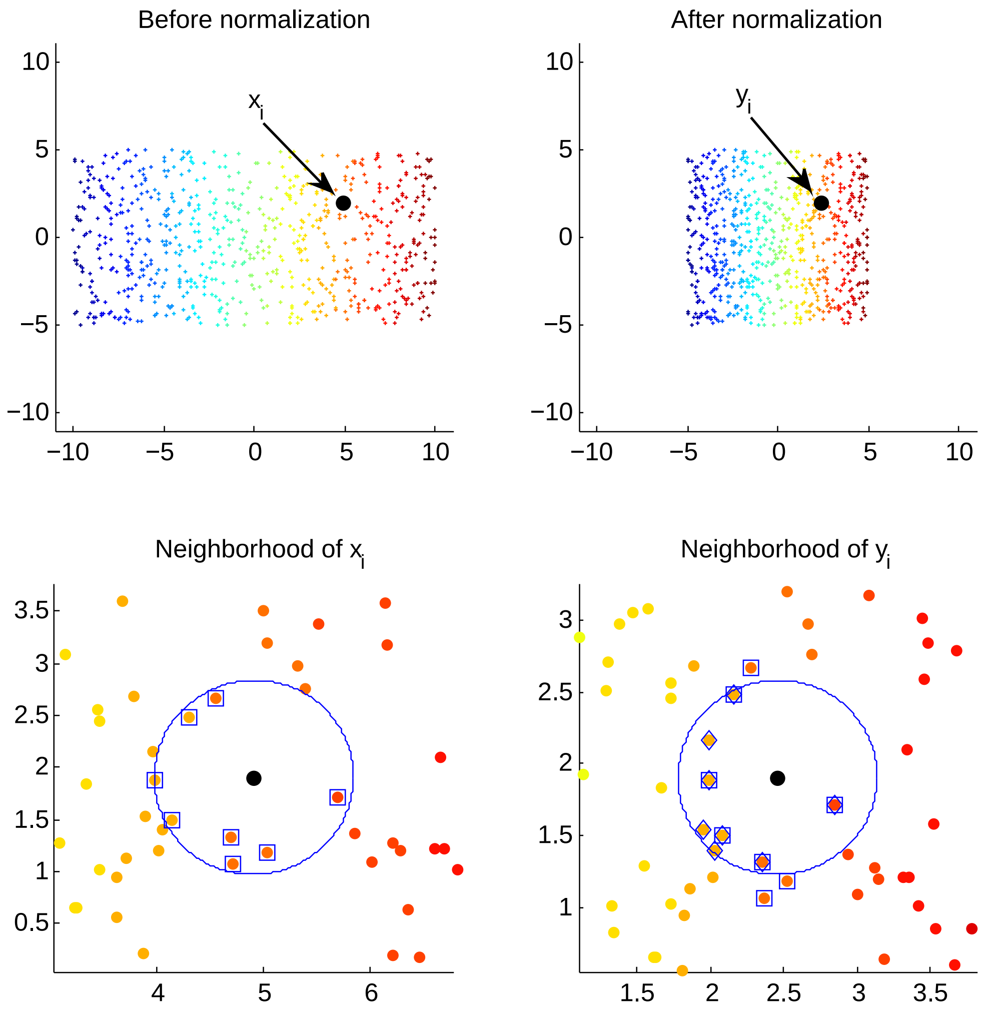

Figure 2.2: Illustration of output normalized methods.

- Output error: The changes of nearest neighbor ranks $\widehat{r}_{ij}$ from the output space to the input space.
- Input error: The changes of nearest neighbor ranks s_{ij} from the input space to the output space.

These two types of errors can be measured by the local rank correlation between the nearest neighbor distances in the input and output spaces. The question is that among all cases in corresponding neighborhoods N_J^I and N_J^O, which cases shall we compare?

One possible way is to consider cases in the union of corresponding neighborhoods. For all j in $\mathcal{N}_J^*(i) = N_J^I \bigcup N_J^O$, define the adjusted ranks

$$S_{ij} = \begin{cases} s_{ij}, & j \in N_J^I(i) \bigcap N_J^O(i) \\ \frac{\gamma+J+1}{2}, & j \notin N_J^I(i) \bigcap N_J^O(i) \end{cases}$$

$$\widehat{R}_{ij} = \begin{cases} \widehat{r}_{ij}, & j \in N_J^I(i) \bigcap N_J^O(i) \\ \frac{\gamma+J+1}{2}, & j \notin N_J^I(i) \bigcap N_J^O(i) \end{cases}$$

where $\gamma = \left| N_J^I(i) \bigcup N_J^O(i) \right|$. Then the topology preservation in the neighborhood of the i-th case is quantified by Spearman's rank correlation within $\mathcal{N}_J^*(i)$,

$$\rho_J(i, \mathbf{X}, \widehat{\mathbf{Y}}) = 1 - \frac{\sum\limits_{j \in \mathcal{N}_J^*(i)} \left\{ \left(S_{ij} - \widehat{R}_{ij} \right)^2 \right\}}{\frac{1}{6}(\gamma^3 - \gamma)} . \tag{2.1}$$

A higher value of $\rho_J(i)$ indicates a better resemblance between the neighborhood $N_J^I(i)$ and $N_J^O(i)$. The drawback about this measure is that it does not behave as a correlation. When the output $\widehat{\mathbf{Y}}$ is generated independently from $\mathbf{X}$, we can show that the expected value of the local Spearman correlation is not zero, which makes its interpretation counter-intuitive.

Another possible choice is to only consider cases in the intersection of corresponding neighborhood. For all j in $N_J^I \bigcap N_J^O$, define the adjusted rank

$$\begin{cases} \delta_{ij} = \left| \left\{ k \in N_J^I \bigcap N_J^O : \|\mathbf{x}_i - \mathbf{x}_k\| \le \|\mathbf{x}_i - \mathbf{x}_j\| \right\} \right| \\ \widehat{\delta}_{ij} = \left| \left\{ k \in N_J^I \bigcap N_J^O : \|\widehat{\mathbf{y}}_i - \widehat{\mathbf{y}}_k\| \le \|\widehat{\mathbf{y}}_i - \widehat{\mathbf{y}}_j\| \right\} \right| \end{cases} \tag{2.2}$$

The topology preservation in the neighborhood of the i-th case is quantified by Kendall's rank correlation within $N_J^I \bigcap N_J^O$,

$$\tau_J(i, \mathbf{X}, \widehat{\mathbf{Y}}) = \begin{cases} \frac{\sum\limits_{j<k \in N_J^I \bigcap N_J^O} \mathrm{sign}\left\{(\delta_{ij} - \delta_{ik})(\widehat{\delta}_{ij} - \widehat{\delta}_{ik})\right\}}{\frac{1}{2}\zeta(\zeta-1)}, & \zeta > 1 \\ 0, & \zeta \le 1 \end{cases} \tag{2.3}$$

where $\zeta = \left| N_J^I \bigcap N_J^O \right|$. This measure does behave as a correlation, i.e. when the output $\widehat{\mathbf{Y}}$ are generated independently from input $\mathbf{X}$, $\mathrm{E}(\tau_J(i)) = 0$ for all i. However, it fails to detect the distortion of topology when outsiders enter the neighborhood while ranks of original neighbors remains the same. As illustrated in Figure 2.3, applying PCA on the V-shaped input data could result in a complete overlap of left and right wings. However, it can be easily seen that the ranking within $N_J^I \bigcap N_J^O$ for all cases does not change and therefore leads to a value of τ_J close to 1, which is against our visual intuition.

These two naive measures are not ideal because measure (2.1) tries to combine the output error and input error, while measure (2.3) considers only the rank discrepancy between $N_J^I(i)$ and $N_J^O(i)$ but ignores the discrepancy in overlapping.

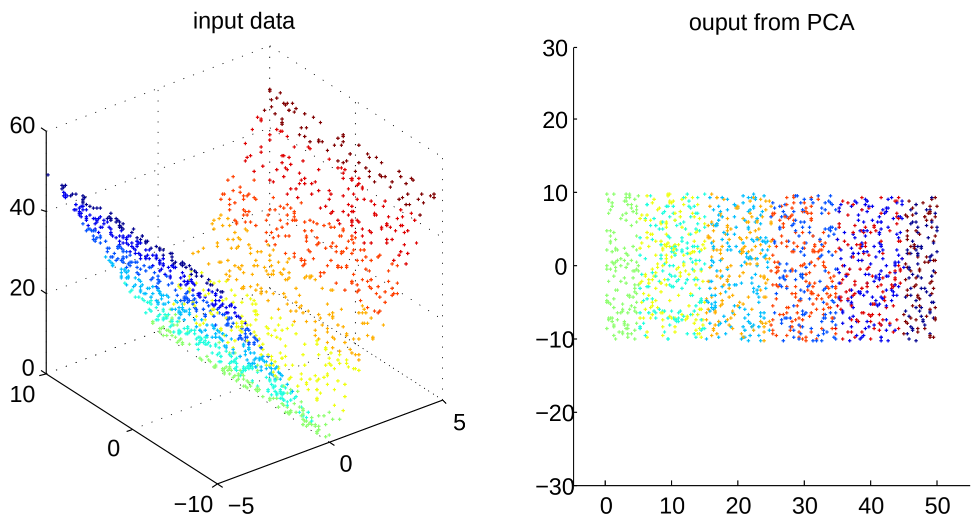

Figure 2.3: Failure of the naive measure.

2.3 Local rank correlation

2.3.1 Definition

In this section, we use local rank correlations to define a family of performance measures that quantify the output and input error separately. For all j in $N_J^I(i) \bigcup N_J^O(i)$, define the adjusted rank

$$S_{ij} = \begin{cases} \delta_{ij}, & j \in N_J^I(i) \bigcap N_J^O(i) \\ \frac{\zeta+J+1}{2}, & j \notin N_J^I(i) \bigcap N_J^O(i) \end{cases} \tag{2.4}$$

$$\widehat{R}_{ij} = \begin{cases} \widehat{\delta}_{ij}, & j \in N_J^I(i) \bigcap N_J^O(i) \\ \frac{\zeta+J+1}{2}, & j \notin N_J^I(i) \bigcap N_J^O(i) \end{cases} \tag{2.5}$$

where $\zeta = \left| N_J^I(i) \bigcap N_J^O(i) \right|$, δ_{ij} and $\widehat{\delta}_{ij}$ are defined in (2.2). The adjustment in (2.4) and (2.5) is to make the ranks comparisons local. Those cases that are not in $N_J^I(i)$ will be considered as rank tied in the input space, and vice versa.

To measure the output error, we can define local rank correlation measures in the neighborhood of each data point in the output space.

Definition 2.1. *Local rank correlation for output error*: Given an input dataset $\mathbf{X}$ and a low-dimensional representation $\widehat{\mathbf{Y}}$, define the local Spearman correlation at the i-th case as

$$\rho_J^O(i, \mathbf{X}, \widehat{\mathbf{Y}}) = 1 - \frac{\sum_{j \in N_J^O(i)} \left\{ \left(S_{ij} - \widehat{r}_{ij} \right)^2 \right\} + U}{\frac{1}{6}(J^3 - J)}, \tag{2.6}$$

where $U = \left[(J-\zeta)^3 - (J-\zeta) \right] / 12$ is the adjustment made for the appearance of ties (Kendall [1948]). We can also define a local Kendall correlation as

$$\tau_J^O(i, \mathbf{X}, \mathbf{Y}) = \frac{\sum_{j<k \in N_J^O(i)} \text{sign} \left\{ [S_{ij} - S_{ik}] \cdot (\widehat{r}_{ij} - \widehat{r}_{ik}) \right\}}{\frac{1}{2} J(J-1)}. \tag{2.7}$$

For a given input dataset $\mathbf{X}$ and a given dimensionality reduction method $\widehat{\psi} : \mathbf{X} \mapsto \widehat{\psi}(\mathbf{X})$, an overall goodness measure can be defined by averaging the local correlation over all cases in the sample.

$$G_J^O(\widehat{\psi}, \mathbf{X}) = \frac{1}{n} \sum_{i=1}^{n} \Gamma_J^O \left(i, \mathbf{X}, \widehat{\psi}(\mathbf{X}) \right), \tag{2.8}$$

where Γ_J^O can be either ρ_J^O, or τ_J^O.

The local rank correlations $\rho_J^O(i)$ or $\tau_J^O(i)$ measure the similarity, in terms of output errors, between the corresponding neighborhoods, $N_J^I(i)$ and $N_J^O(i)$. Similarly, we can define local rank correlations to measure the input error.

Definition 2.2. *Local rank correlation for input error*: Given an input dataset $\mathbf{X}$ and a low-dimensional representation $\widehat{\mathbf{Y}}$, the local Spearman correlation and local Kendall correaltion for the input error at the i-th case are defined as

$$\rho^I_{_J}(i, \mathbf{X}, \widehat{\mathbf{Y}}) = 1 - \frac{\sum\limits_{j \in N^I_J(i)} \left\{ \left(s_{_{ij}} - \widehat{R}_{_{ij}} \right)^2 \right\} + U}{\frac{1}{6}(J^3 - J)}\,, \tag{2.9}$$

$$\tau^I_{_J}(i, \mathbf{X}, \widehat{\mathbf{Y}}) = \frac{\sum\limits_{j<k \in N^I_J(i)} \operatorname{sign} \left\{ [\widehat{R}_{_{ij}} - \widehat{R}_{_{ik}}] \cdot (s_{_{ij}} - s_{_{ik}}) \right\}}{\frac{1}{2}J(J-1)}\,. \tag{2.10}$$

The overall goodness measure of a given method $\widehat{\psi}$ and input data $\mathbf{X}$ is defined as

$$G^I_{_J}(\widehat{\psi}, \mathbf{X}) = \frac{1}{n} \sum_{i=1}^{n} \Gamma^I_J \left(i, \mathbf{X}, \widehat{\psi}(\mathbf{X}) \right)\,, \tag{2.11}$$

where Γ^I_J can be either $\rho^I_{_J}$, or $\tau^I_{_J}$.

2.3.2 Remark

The proposed local rank correlations have some nice properties. The higher values of local measures $\Gamma^I_{_J}(i)$ and $\Gamma^O_{_J}(i)$ indicate a higher degree of similarity between the original data and the low-dimensional configuration in the neighborhood of case i, while values close to 0, or negative values indicate that low-dimensional configuration fails to preserve the local structure of the input data in certain neighborhoods. Two special situations are:

- $\Gamma^I_{_J}(i) = \Gamma^O_{_J}(i) = 1$ if all the ranking relationships of the observed data $\mathbf{X}$ in the neighborhood of case i are preserved exactly in the corresponding neighborhood in the output data $\widehat{\mathbf{Y}}$.
- The expected values $\mathrm{E}\left[\Gamma^I_{_J}(i) \right]$ and $\mathrm{E}\left[\Gamma^O_{_J}(i) \right]$ are both zero, for any case i, where the output $\widehat{\mathbf{Y}}$ is generated by an algorithm which is stochastically independent of the input data $\mathbf{X}$.

These two facts hold for both local Spearman and Kendall correlations. Notice that the second situation is worse than we can have in practice. Moreover, the local measures $\Gamma_J^I(i)$ and $\Gamma_J^O(i)$, can achieve negative values for some i. Nevertheless, the overall goodness measures G_J^O and G_J^I, for sensible algorithms, will take values between 0 and 1. Note that, the ranks and rank correlations are being used to achieve robustness against the presence of outliers. The distributions of both local Spearman and Kendall correlations when $\mathbf{X}$ and $\widehat{\mathbf{Y}}$ are independent are derived in Appendix B.

The computational complexity is also of interest. To calculate the goodness measure, we first construct the J-nearest neighbor graph for both $\mathbf{X}$ and $\widehat{\mathbf{Y}}$. This step scales as $O(n^2p)$. In the next step, we calculate the local rank correlation in each neighborhood. This scales (in each neighborhood) as $O(J)$ for Spearman ρ_J and $O(J \log J)$ for Kendall τ_J. Therefore, since $J \leq n$, the total complexity of calculating G_J^I (or G_J^O) for ρ_J scales as $O(n^2p)$. The total complexity of calculating G_J^I (or G_J^O) for τ_J scales as $O(n^2p + nJ \log J)$.

To use the proposed goodness measure G_J for assessing the performance of a dimension reduction method, four local measures can be chosen. We may choose either Γ_J^I or Γ_J^O, and we may also choose to use either Spearman ρ_J or Kendall τ_J. The measures Γ_J^I and Γ_J^O quantify different types of errors in dimension reduction. Although these two types of errors usually occur together, having both G_J^I and G_J^O provide more complete information about the performance of a given method.

2.3.3 Choice of J

In the proposed measures, J is a user-specified tuning parameter, which specifies the neighborhood size for local rank comparisons. Notice that some nonlinear dimensionality reduction methods start with a K-nearest neighbors graph, and K is also a user-specified parameter. The choice of J in the local rank correlation does not have to depend on the value of K. Ideally, J needs to be selected small enough that in each neighborhood the underlying manifold is approximately Euclidean. One strategy to choose J is to plot J versus $G_{_J}(\widehat{\psi}, \mathbf{X})$ as shown for example in Figure 2.7 and Figure 2.8. A value of J which is chosen from an interval over which $G_{_J}(\widehat{\psi}, \mathbf{X})$ is stable, is a reasonable candidate for the algorithm.

2.3.4 Adjustments for output-normalized methods

As mentioned in Section 2.1.2, normalizing the output $\widehat{\mathbf{Y}}$ of a dimension reduction method will distort the structure of neighborhoods, so that the topological structure will not typically be preserved by the output configuration. It is not adequate to check the topology preservation between $\mathbf{X}$ and $\widehat{\psi}(\mathbf{X})$ for those output-normalized methods. Instead, we will look for a transformation matrix $\widehat{\mathbf{A}}_{q\times q}$, and assess the performances of output-normalized methods by an adjusted measure

$$G_J^{\mathbf{A}}(\widehat{\psi}, \mathbf{X}) = \frac{1}{n}\sum_{i=1}^{n} \Gamma_{_J}(i, \mathbf{X}, \widehat{\psi}_{\mathbf{A}}(\mathbf{X}))\,, \tag{2.12}$$

where $\widehat{\psi}_{\mathbf{A}}(\mathbf{X}) = \widehat{\psi}(\mathbf{X}) \cdot \widehat{\mathbf{A}}$, and $\Gamma_{_J}$ can be $\rho^I_{_J}$, $\rho^O_{_J}$ or $\tau^I_{_J}$, $\tau^O_{_J}$.

It is hoped that after the affine transformation $\widehat{\mathbf{A}}$, $\widehat{\psi}_{\mathbf{A}}(\mathbf{X})$ can preserve the proximities between neighboring points as much as possible, i.e. $\widehat{\mathbf{A}}$ will minimize the least squared error

$$\sum_{i}^{n} \sum_{j\in N^I_{_J}(i)} \left[(\mathbf{x}_i - \mathbf{x}_j)'(\mathbf{x}_i - \mathbf{x}_j) - (\mathbf{y}_i - \mathbf{y}_j)'\mathbf{A}'\mathbf{A}(\mathbf{y}_i - \mathbf{y}_j)\right]^2 \,,$$

where $\mathbf{x}_i$ and $\mathbf{y}_i$ be the corresponding point in the original data and in the output of the algorithm $\widehat{\psi}$, respectively.

However, to make $G_{_J}^{\mathbf{A}}(\widehat{\psi}, \mathbf{X})$ robust against outlying points, we will find the transformation matrix $\widehat{\mathbf{A}}$ that minimizes the least squared error over an outlier free subset $\mathbb{I}$, so that

$$\widehat{\mathbf{A}} = \underset{\mathbf{A} \in \mathbb{C}^q}{\arg\min} \sum_{i \in \mathbb{I}} \sum_{j \in N_{_J}^I(i)} \left[(\mathbf{x}_i - \mathbf{x}_j)'(\mathbf{x}_i - \mathbf{x}_j) - (\mathbf{y}_i - \mathbf{y}_j)'\mathbf{A}'\mathbf{A}(\mathbf{y}_i - \mathbf{y}_j) \right]^2 . \tag{2.13}$$

The detailed procedure to select the subset $\mathbb{I}$ and solve equation (2.13) is provided in Appendix A. The complexity of the procedure scales as $O(n^2)$.

2.4 Numerical experiments

In this section, we conduct numerical examples on three benchmark datasets to illustrate the usefulness of the local rank correlation.

Example 2.1. *The Swiss roll and the S-curve:* In this experiment, $n = 1000$ data points are generated randomly from two manifolds, the Swiss roll and the S-curve. They are both 2-dimensional manifolds embedded into $\mathbb{R}^3$ (Figure 2.4). The data points are colored to help readers recognize the structure of the manifolds. Among many dimension reduction methods, we choose four diverse methods, namely ISOMAP, LTSA, MVU, and PCA. Figure 2.5 and Figure 2.6 show four output configurations in $\mathbb{R}^2$ from these methods for the Swiss roll and the S-curve, respectively. We evaluate the performance of four methods by the local rank correlations $G_{_J}^I(\widehat{\psi}, \mathbf{X})$ and $G_{_J}^O(\widehat{\psi}, \mathbf{X})$ with both Spearman $\rho_{_J}$ and Kendall $\tau_{_J}$. Notice that LTSA is an output-normalized method, and therefore, its performance is assessed by the adjusted measures (2.12). All details about tuning and computation can be found in Appendix A.

The goodness measures are calculated under different values of J. Figure 2.7 and Figure

2.8 show the values of $G_J(\widehat{\psi}, \mathbf{X})$ for each method as functions of J. Figure 2.9 and Figure 2.10 show the histogram of ρ_J^O and ρ_J^I in the Swiss roll data.

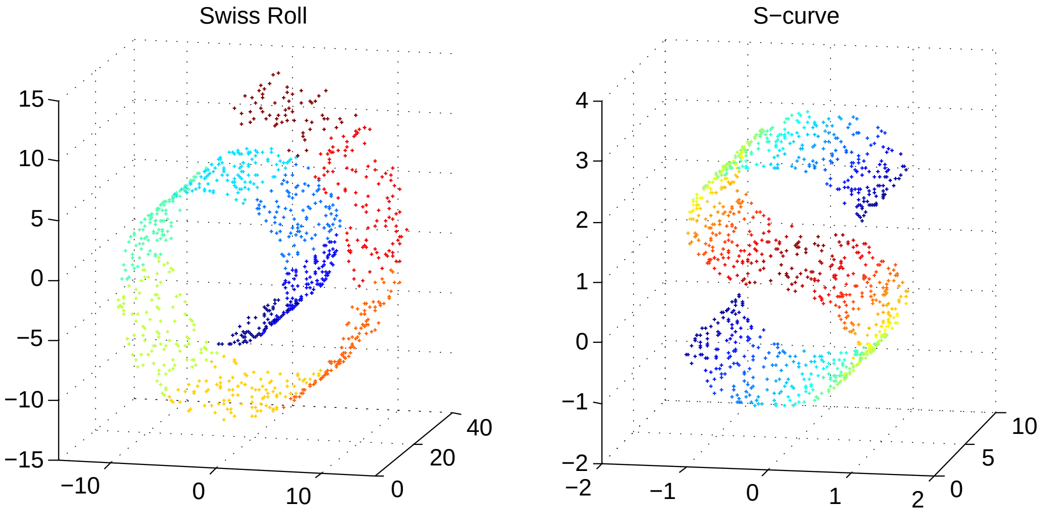

Figure 2.4: The Swiss roll and the S-curve

As can be seen from Figure 2.5 and Figure 2.6, in the 2-dimensional configurations from PCA in both the Swiss roll and the S-curve, the points with different colors are mixed together, because PCA fails to recover the nonlinear structure of the embedded data. Among three nonlinear methods, the configurations from MVU preserve the structure to some extent. Points with different colors are reasonably separated in the middle, but they mix a little at boundaries. Both LTSA and ISOMAP preserve the color level well, indicating a better embedding than MVU and PCA. These facts are all correctly reflected by the four goodness measures $G_J(\widehat{\psi}, \mathbf{X})$ in Figure 2.7 and Figure 2.8. Also, both figures show that all the four measures are stable within a reasonable range of J.

We also compare the local rank correlation (LRC) with the goodness measures, MRREs, T&C, and LCMC (all with $J = 6$) described in Section 2.1.1. The results are reported in

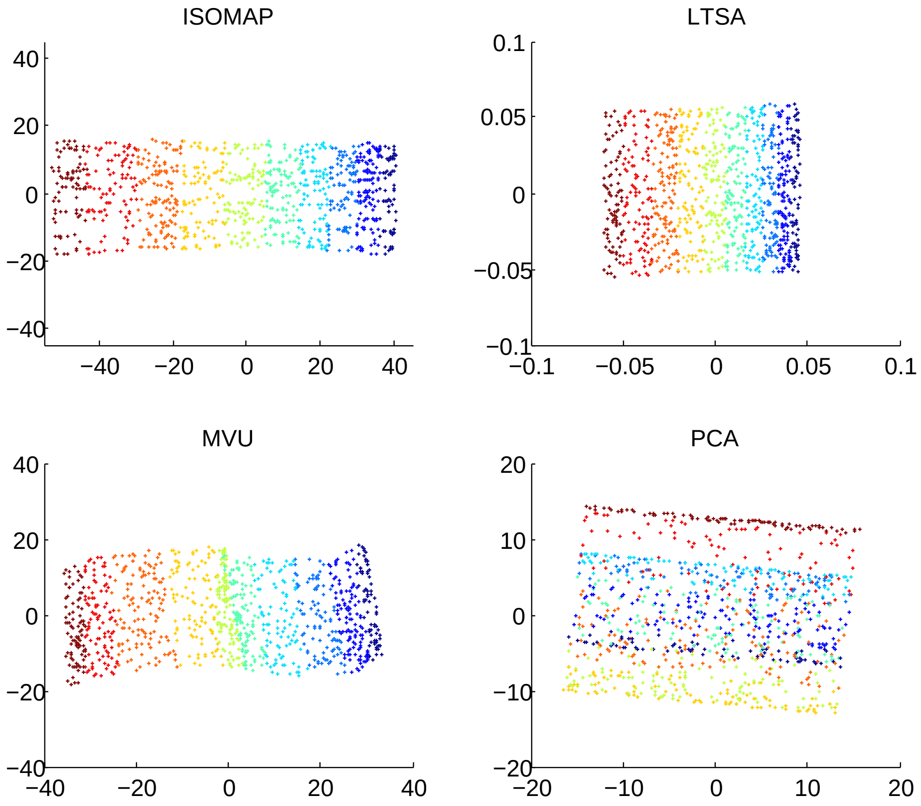

Figure 2.5: Two-dimensional output configurations of different methods for the Swiss roll

Table 2.1 and Table 2.2.

We can see from Table 2.1 and Table 2.2, MRREs and T& C tend to have high values for all methods with little separation. This makes them difficult for users to interpret. The measure LCMC has better separation on different algorithms. However, in the Swiss roll (Table 2.1), its value cannot correctly reflect the performance of LTSA, because the output of LTSA has a normalization constraint, which only preserves the neighborhood geometry of the input data up to an affine transformation.

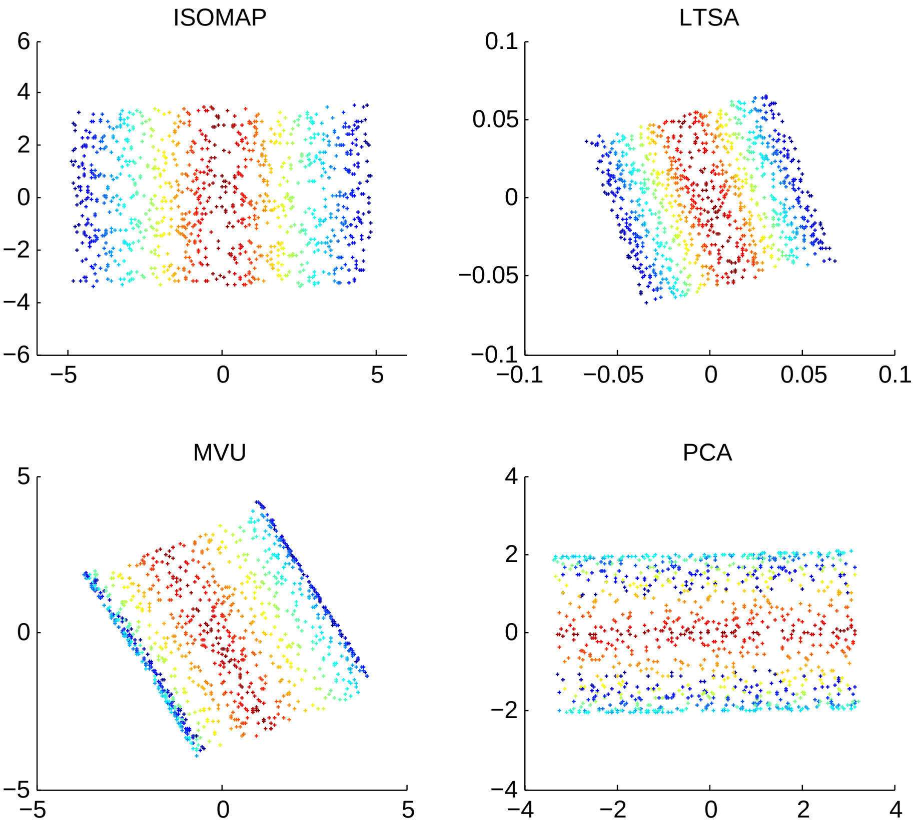

Figure 2.6: Low-dimensional configurations of different methods for S-curve

Example 2.2. *Sculpture face images:* The sculpture face dataset (Tenenbaum et al. [2000]) includes 698 images, each image having 64 × 64 pixels of a sculpture face while varying three free parameters: left-right pose, up-down pose, and lighting direction. So the data are originally in $\mathbb{R}^{64\times 64}$. We apply ISOMAP, LTSA, MVU, and PCA on the data to obtain 2-dimensional representations. Figure 2.11 shows 2-dimensional configurations from these four different methods. In the figure, each point represents an image and we have selected 12 images for display. A common pattern appears in all three nonlinear methods (ISOMAP, LTSA, MVU), namely that the horizontal axis roughly represents the left-right pose, and

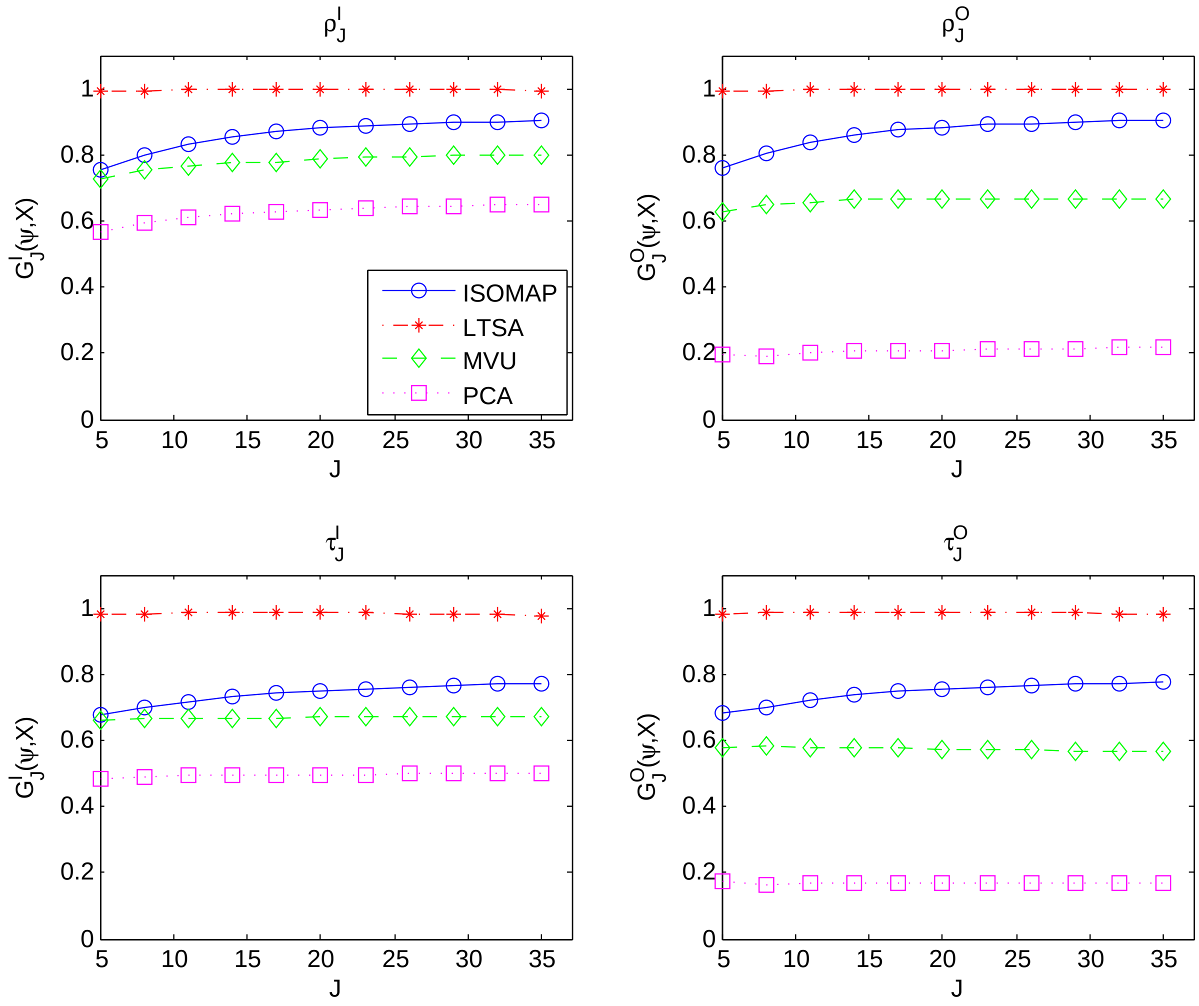

Figure 2.7: Local Spearman correlation as functions of J in the Swiss roll

the vertical axis represents the up-down pose. The linear method PCA does not show any clear pattern.

We use local rank correlations, with both ρ_J and τ_J, to assess the performance of the four different methods. Figure 2.12 shows the goodness measures as functions of J. The comparisons between local rank correlation, MRREs, T& C, and LCMC (all with

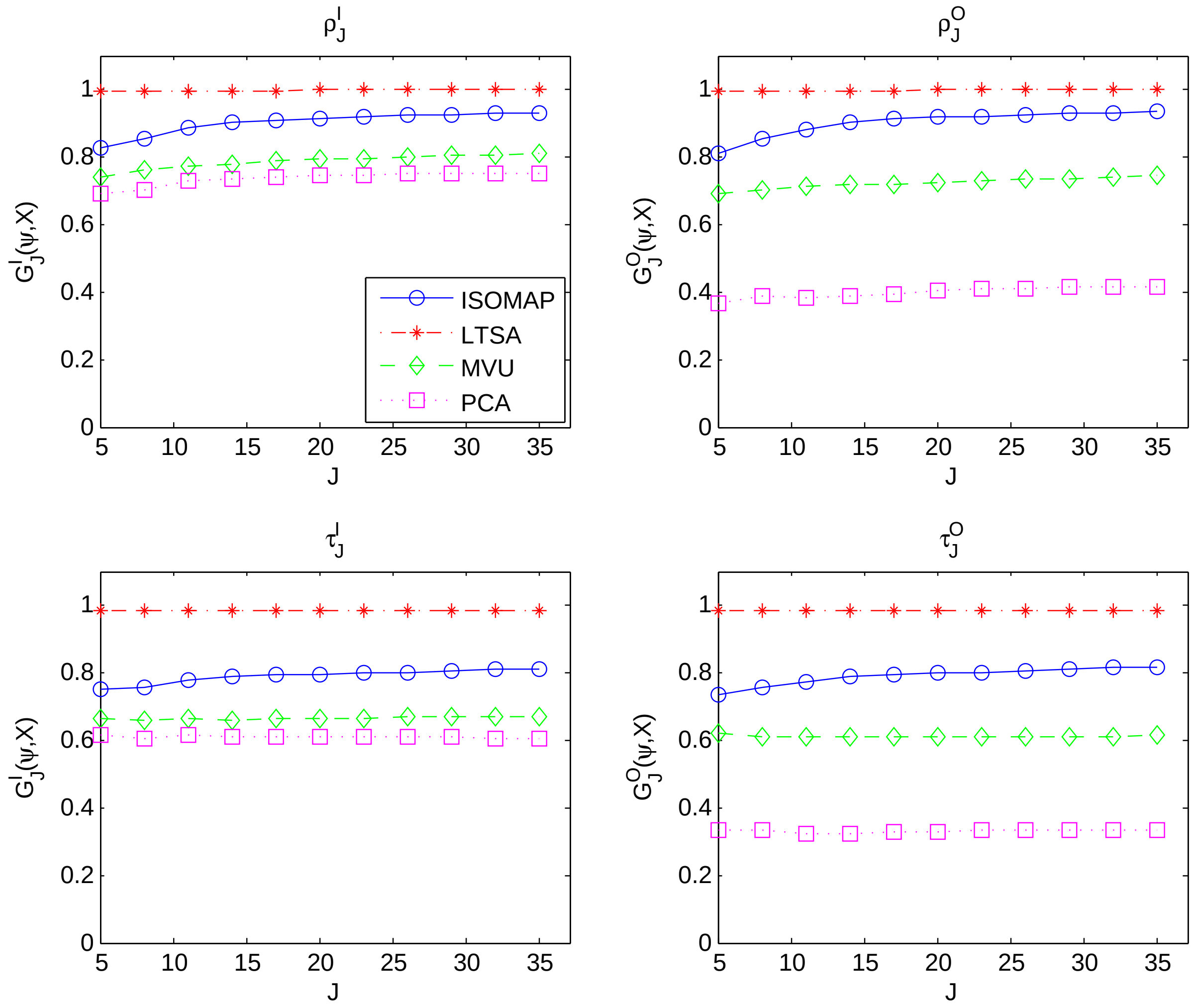

Figure 2.8: Local Spearman correlation as functions of J in the S-curve

$J = 6$), are summarized in Table 2.3. As can be seen, local rank correlation suggests that ISOMAP and MVU in this dataset outperform LTSA, and all three nonlinear methods, ISOMAP, MVU and LTSA, outperform the linear method PCA. This result coincides with my personal visual intuition from Figure 2.11.

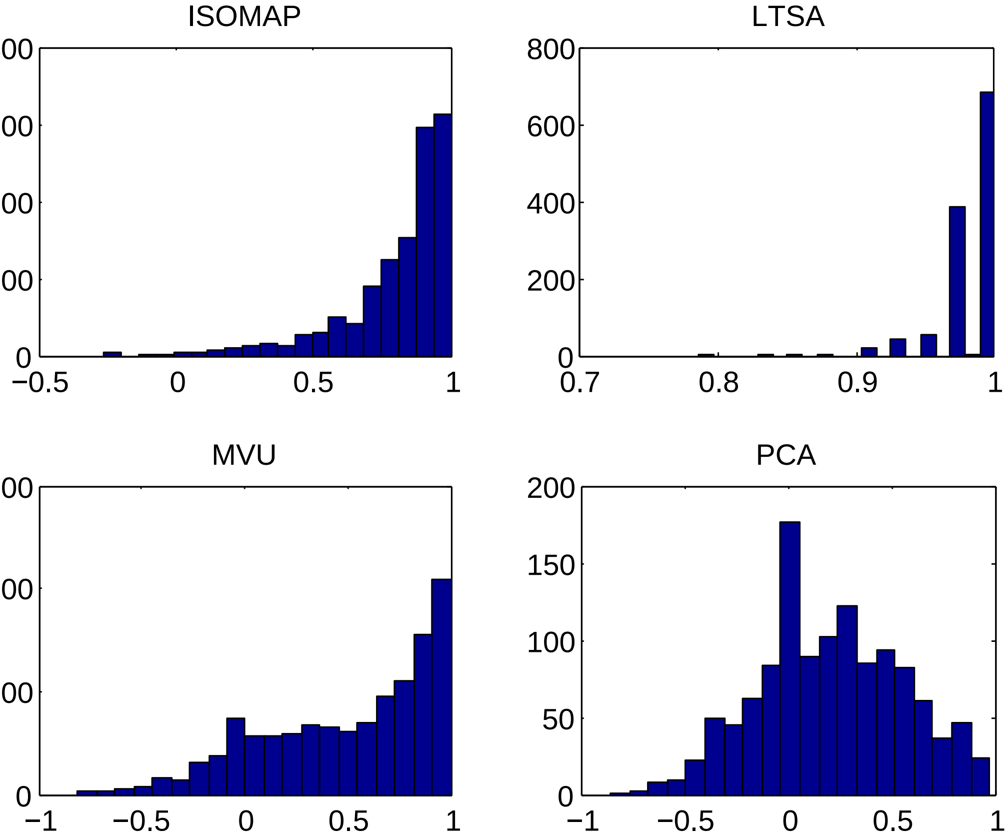

Figure 2.9: Histogram of ρ_J^O $(J = 6)$ in the Swiss roll

2.5 Choosing tuning parameters for algorithms

In addition to assessing the performance of dimension reduction methods, local rank correlations can be used in some other issues in dimension reduction. In this section, we discuss how can we use local rank correlations to help choosing tuning parameters for dimension reduction algorithms. Many nonlinear dimension reduction methods start with constructing the K-nearest neighbor graph, and the neighborhood size K is usually a tuning parameter

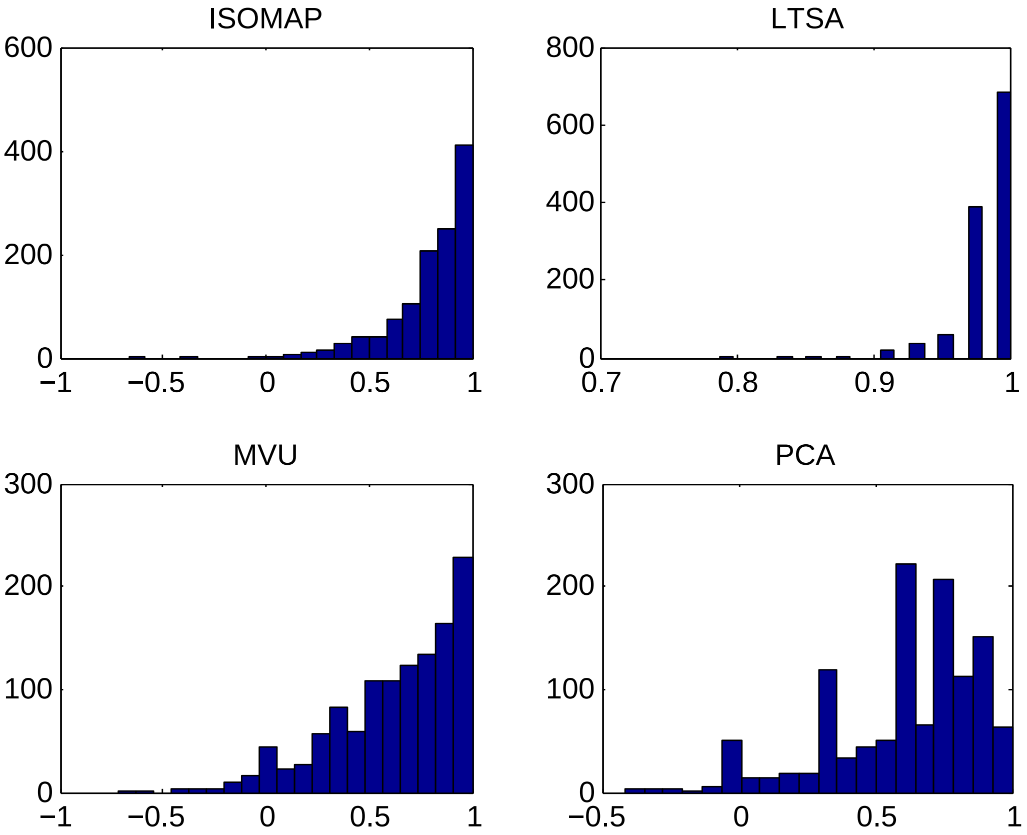

Figure 2.10: Histogram of ρ_J^I $(J = 6)$ in the Swiss roll

in the algorithm. The success of graph-based nonlinear dimensionality reduction methods depends heavily on the selection of K. If K is chosen to be too small, the local geometric structure cannot be accurately represented in the neighborhood graph. On the other hand, if K is chosen to be too large, the K-nearest neighbor graph will contain *shortcuts*, i.e. two points will be mistakenly considered as neighbors when they are in fact far away on the manifold (See Figure 2.13).

This problem often appears in noisy data, and will cause a serious damage on the

Methods	LRC				MRREs		T & C		LCMC
	ρ^{I}_{J}	ρ^{O}_{J}	τ^{I}_{J}	τ^{O}_{J}	M_O	M_I	T	C	
ISOMAP	0.787	0.782	0.701	0.698	0.999	0.999	0.999	0.999	0.894
LTSA	0.988	0.978	0.981	0.975	0.999	0.998	0.993	0.998	0.609
MVU	0.703	0.623	0.653	0.578	0.999	0.999	0.996	0.999	0.828
PCA	0.594	0.198	0.483	0.171	0.998	0.997	0.883	0.995	0.415

Table 2.1: Assessing ISOMAP, LTSA, MVU, PCA in Swiss Roll data ($J = 6$)

Methods	LRC				MRREs		T & C		LCMC
	ρ^{I}_{J}	ρ^{O}_{J}	τ^{I}_{J}	τ^{O}_{J}	M_O	M_I	T	C	
ISOMAP	0.816	0.804	0.763	0.803	0.999	0.999	1.000	1.000	0.891
LTSA	0.994	0.993	0.983	0.979	0.999	0.999	0.999	0.999	0.867
MVU	0.721	0.646	0.695	0.617	0.999	0.998	0.993	0.997	0.754
PCA	0.673	0.375	0.388	0.369	0.998	0.998	0.963	0.998	0.584

Table 2.2: Assessing ISOMAP, LTSA, MVU, PCA in S-curve data ($J = 6$)

performance of graph-based methods Saul and Roweis [2003]; Chen and Buja [2009]. Several methods have been proposed in order to select the optimal value of K. For example Shao and Wan [2012] also proposed a strategy to detect the appearance of the *shortcuts* via local PCA reconstruction error, and used the Bayesian Information Criterion to obtain the optimal value of K. Another method is proposed in Zhang et al. [2012] that it adaptively selects the neighborhood size K_i for each point. In this method, the manifold is parameterized and then the first-order Taylor expansion is applied at each input data point to analyze the relationship between neighboring points, based on which a criterion is

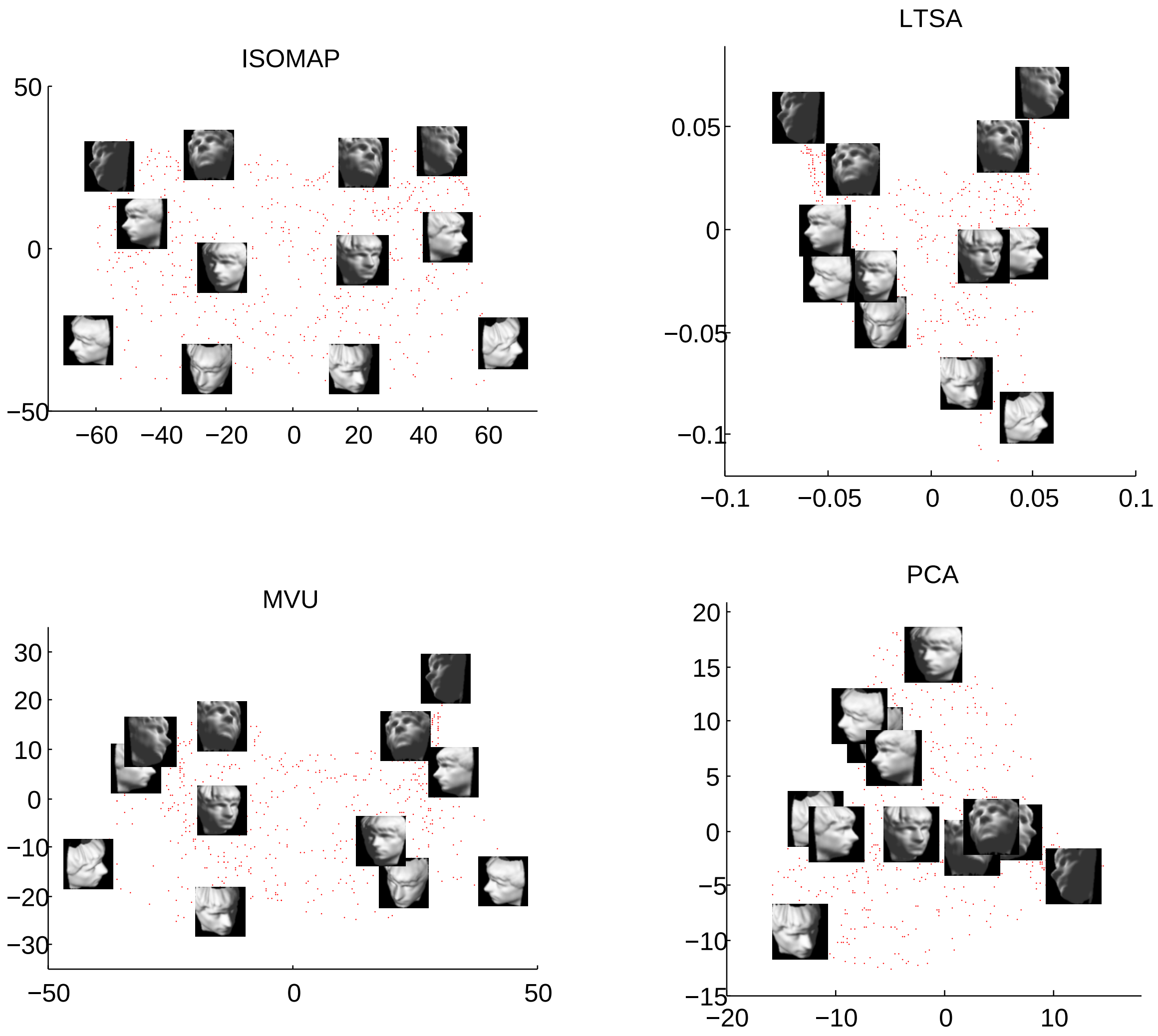

Figure 2.11: 2-dimensional configurations from ISOMAP, LTSA, and PCA for sculpture face image

defined to help in identifying the neighborhood. In Pavan and Pelillo [2007] and Yang and Latecki [2011], authors introduced an additional sparsification parameter and used clustering techniques to select the dominant subset of the K-nearest neighbors. Other related work include Shao et al. [2007], Mekuz and Tsotsos [2006], Premachandran and Kakarala

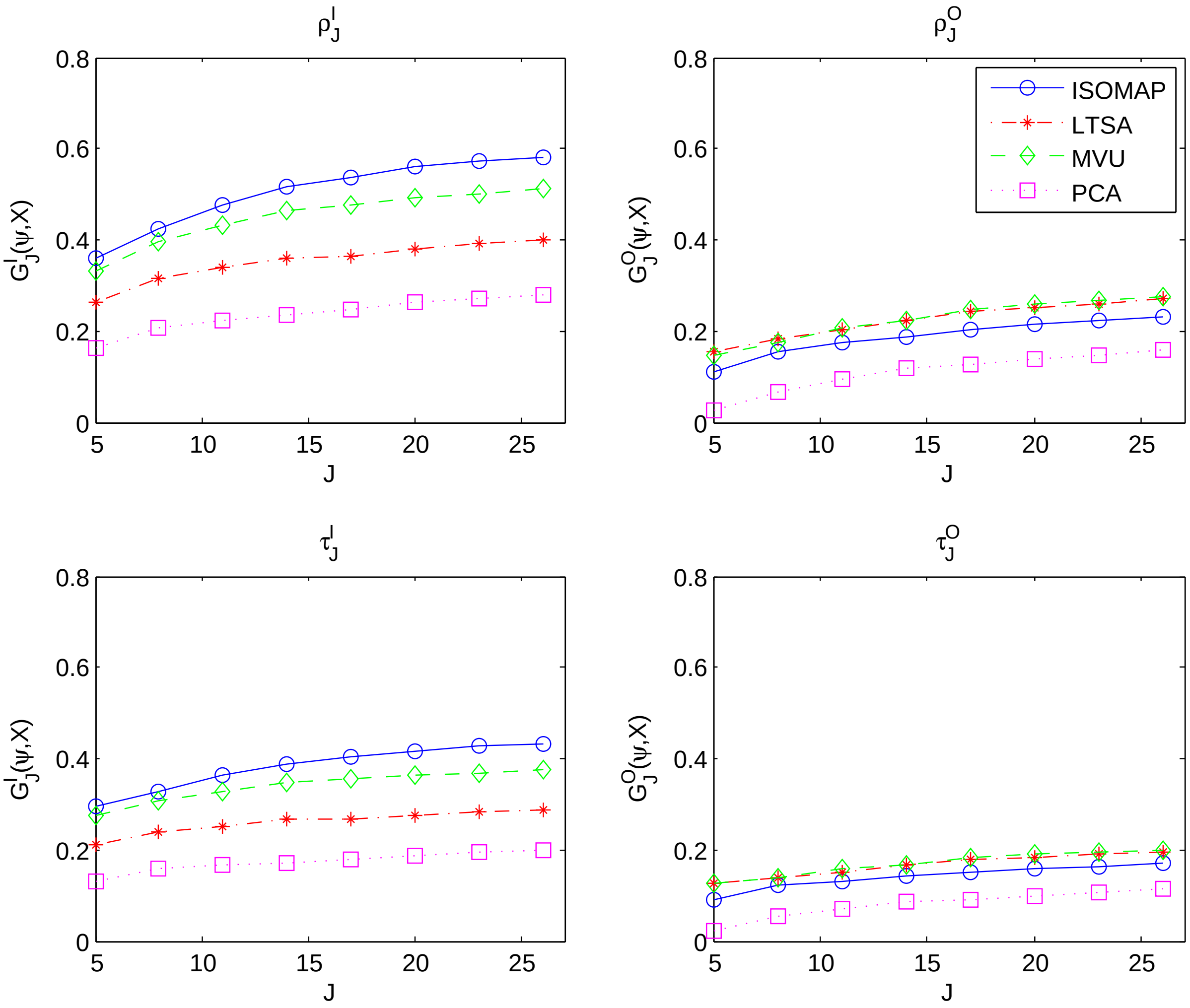

Figure 2.12: Local rank correlation as functions of J in sculpture face image

[2013] and Kouropteva et al. [2002]. However, there is yet no standard way of selecting the optimal value of K and in practice, K is usually chosen by experience or trial and error.

The local rank correlation provides us a reliable criterion in choosing K. For a given input dataset and a dimension reduction algorithm, we may apply the algorithm over a range of values of K, and calculate $G_J(\widehat{\psi}, \mathbf{X})$ as a function of K (as shown in Figure 2.15).

Methods	LRC				MRREs		T & C		LCMC
	ρ^I_J	ρ^O_J	τ^I_J	τ^O_J	M_O	M_I	T	C	
ISOMAP	0.379	0.116	0.301	0.096	0.997	0.995	0.825	0.991	0.394
LTSA	0.284	0.158	0.208	0.128	0.997	0.993	0.898	0.979	0.348
MVU	0.338	0.149	0.287	0.128	0.997	0.994	0.847	0.987	0.391
PCA	0.189	0.028	0.162	0.017	0.997	0.989	0.905	0.960	0.233

Table 2.3: Assessing ISOMAP, LTSA, MVU, PCA in sculpture face image data ($J = 6$)

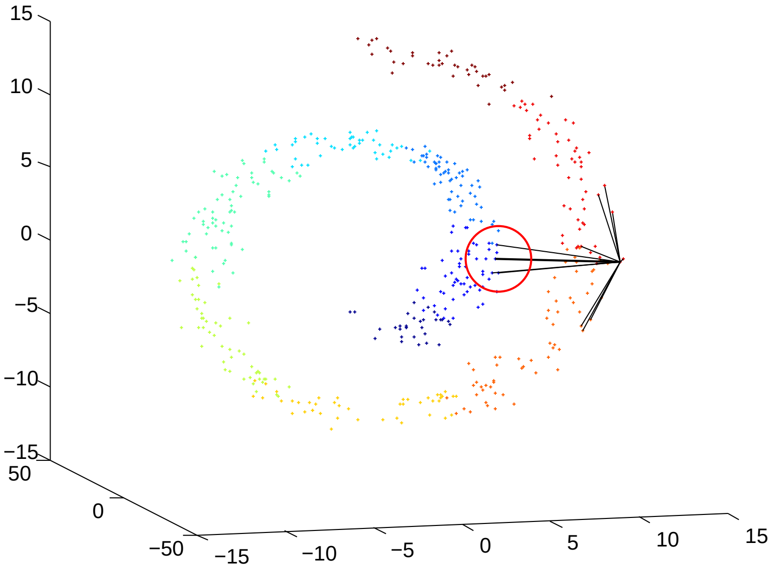

Figure 2.13: Presence of *shortcuts* in K-NN graph

Since $G_J(\widehat{\psi}, \mathbf{X})$ measures the performance of $\widehat{\psi}$, we can pick the K that corresponds to the largest $G_J(\widehat{\psi}, \mathbf{X})$.

Example 2.3. *Selecting neighborhood size K in ISOMAP:* Here we consider the performance of ISOMAP on the Swiss roll manifold. We demonstrate that it is risky to make a desultory choice of K, and how local rank correlation can solve this problem.

The data are generated randomly on the Swiss roll manifold with sample size $n = 1500$. The ISOMAP algorithm is applied on the data with different values of K, and Figure 2.14 shows the respective low-dimensional configurations. In each case, the performance is evaluated by the local rank correlation and displayed as a function of K in Figure 2.15.

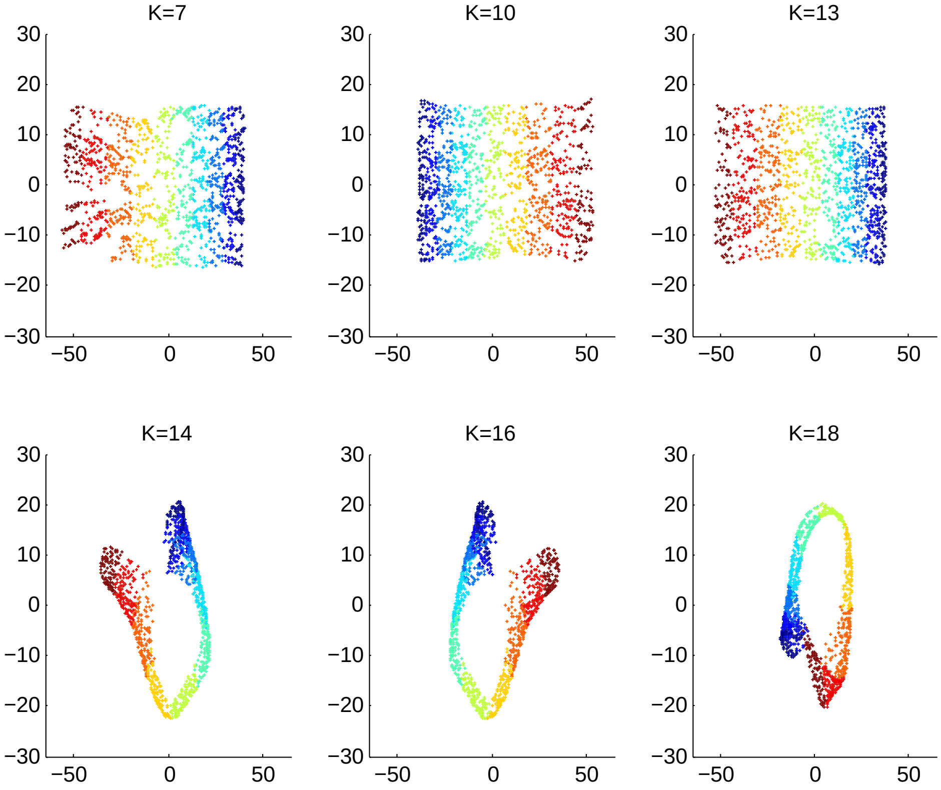

Figure 2.14: Low-dimensional configurations with different values of K

In Figure 2.15, the left panel shows G_J^I and G_J^O with Spearman ρ_J, and the right panel shows G_J^I and G_J^O with Kendall τ_J. As can be easily noticed in Figure 2.14, the performance of ISOMAP gets better as K increases from $K = 7$ to $K = 13$. A crucial change has

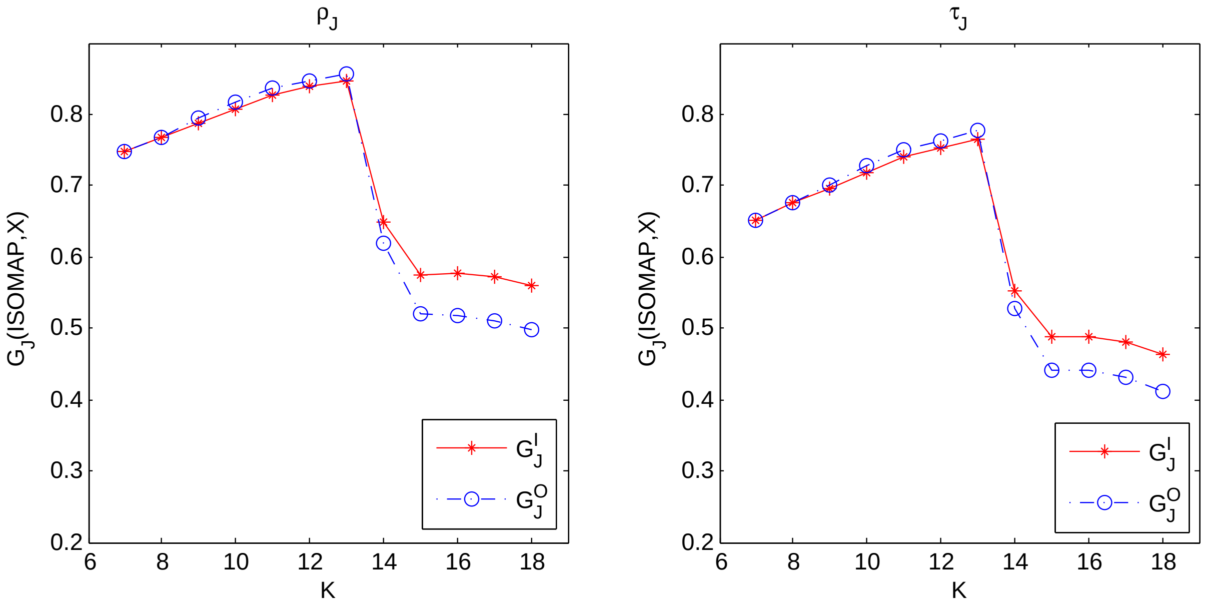

Figure 2.15: Local rank correlation as a function of K ($J = 6$)

happened at points $K = 13$ and $K = 14$. In these two situations, the neighborhood sizes only differ by 1 but the corresponding configurations suddenly become unsatisfactory (at $K = 14$). The fact is correctly captured by the local rank correlation and reflected in Figure 2.15. In all four measures, we observe a peak at $K = 13$, and a steep drop at $K = 14$.

The phenomenon is certainly not limited to this example and to ISOMAP. For nonlinear methods which contain the neighborhood size K as a tuning parameter, it is often desirable to choose a relatively large value of K to get a better embedding. On the other hand, a too large K will invalidate the procedure. The local rank correlations can be good criteria for users to select a good value of the parameter K. If an algorithm contains other tuning parameters, this idea can be also applied.

2.6 Estimating the intrinsic dimensionality of a manifold

Another key parameter in dimension reduction algorithms is the intrinsic dimensionality q. Previous work on intrinsic dimensionality estimation includes two main categories, eigenvalue methods and geometric methods. Eigenvalue methods are based on PCA or the nonlinear generalization of PCA. The intrinsic dimension is determined by thresholding the eigenvalues. Methods in this category includes Fukunaga and Olsen [1971]; Bruske and Sommer [1998]; Verveer and Duin [1995]. Geometric methods are based on fractal dimensions. Methods in this category includes Levina and Bickel [2004]; Grassberger and Procaccia [2004]; Camastra and Vinciarelli [2002]; Kégl [2002]. A detailed review can be found in Lee and Verleysen [2007].

The local rank correlation can be also applied to help in estimating the intrinsic dimensionality. The idea is that if the dimensionality of the low-dimensional representation is chosen to be too small, important features of the original data might be "collapsed" onto the same dimensions so that the topological structure cannot be preserved very well. In this case, the local rank correlation, as a function of q, should rapidly increase as q increases. On the other hand, when all important dimensions have been chosen, the remaining dimensions are assumed to contain only noise. Therefore the local rank correlation would become stable once we achieve a sufficiently large q.

In practice, for a given dataset $\mathbf{X}$ and a chosen method $\widehat{\psi}$, one may apply the method with different values of q, and evaluate the performances of $\widehat{\psi}$ by $G_J(\widehat{\psi}, \mathbf{X})$. We estimate the intrinsic dimensionality by plotting $G_J(\widehat{\psi}, \mathbf{X})$ as a function of q, and choosing the value q, beyond which $G_J(\widehat{\psi}, \mathbf{X})$ becomes stable.

Example 2.4. *Estimating the intrinsic dimensionality of the sculpture face data*: The sculpture face images are recorded as 64×64 vectors. Since images are taken from the same sculpture face by varying three parameters, i.e. left-right pose, up-down pose, and lighting direction, the intrinsic dimensionality of the manifold on which these data vectors lie is three. We apply the ISOMAP algorithm with different values of q having chosen the neighborhood size $K = 8$. The local rank correlations are calculated as functions of q. In Figure 2.16, the left panel shows G_J^I and G_J^O with Spearman ρ_J, and the right panel shows G_J^I and G_J^O with Kendall τ_J. As can be seen, all four curves become stable beyond $q = 3$, based on which we estimate the intrinsic dimensionality to be $\widehat{q} = 3$. Note that in this example, the choice of the tuning parameter K in ISOMAP clearly affects the estimate. For this procedure to work well, a reasonably good choice of K is needed.

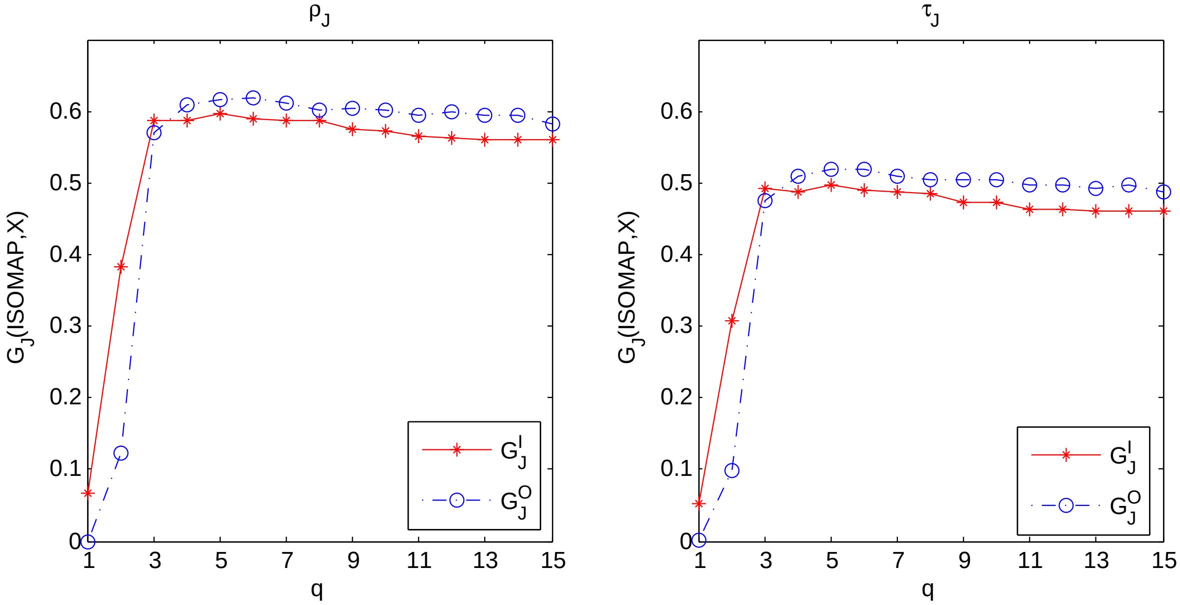

Figure 2.16: Local rank correlation as a function of dimensionality q ($J = 6$)

2.7 Discussion and future work

In this chapter, we developed a local rank correlation measure which quantifies the performance of dimension reduction methods by assessing the preservation of the topology in low-dimensional representations. An adjustment is available so that the measure can correctly assess the performance of the output-normalized methods. The local rank correlation is easily interpretable, and robust against the presence of outliers. The distribution of both local Spearman and Kendall correlation when $\mathbf{X}$ and $\widehat{\mathbf{Y}}$ are independent were developed. A potential future research topic is to further investigate properties of local rank correlations. In addition to the chosen method, values of local rank correlations generally depend on the dimension p and the intrinsic dimension q of $\mathbf{X}$, and also depend on how noisy $\mathbf{X}$ are (can be represented by the variance of random error σ). Two extreme values 0 and 1 are not always achievable in any case. It is of our interest to get upper or lower bounds of local rank correlations given certain p, q and σ, and that would be a benchmark for us to better interpret values of local rank correlations. Besides, a more challenging task is to derive general distributions of local rank correlations, based on which we can develop a goodness-of-fit test of dimension reduction methods.

Chapter 3

Sensitivity Analysis in Dimension Reduction

3.1 Review of related work

Generally, the purpose of the sensitivity analysis in the study of robustness is two-fold. The first goal is to understand the robustness of a statistical method, and the second goal is to flag potential outliers. Within the dimensional reduction framework, the sensitivity analysis first appeared in the study of principal component analysis. The major tool in studying the robustness of PCA is the influence function introduced by Hampel [1968]. The influence function of PCA has been developed from two different perspectives. Since PCA and its robust variants are usually based on the eigendecomposition of some scatter matrix, the robustness of PCA can be measured via the robustness of scatter matrix estimators. The first type of influence functions for PCA (or variants of PCA) are defined on top q eigenvalues λ_i and eigenvectors $\mathbf{e}_i$ of the corresponding scatter matrix estimator.

Critchley [1985] derived influence functions for eigenvalues and eigenvectors of sample covariance matrix, and treated them as influence measures for traditional PCA. Three sample versions of influence measures are also given in that paper. Influence functions for eigenvalues and eigenvectors of the sample correlation matrix have been derived in Calder [1986], and considered as influence measures of PCA. Influence functions of robust PCA based on robust scatter matrix estimators have also been developed, including Huber [1981] for M-estimator, Lopuhaa [1989, 1999] for S-estimator and Croux and Haesbroeck [1999] for MCD-estimator. Croux and Haesbroeck [2000] provided a general definition of influence functions for any robust PCA based on some consistent and affine equivariant scatter matrix estimate. Huang et al. [2009] and Debruyne et al. [2010] generalized this type of influence functions into Kernel PCA framework. Notice that in all aforementioned work the influence function of each eigenvalue is defined as a real-valued function, and of each eigenvector is defined as a vector-valued function. A main drawback of this type of influence functions is that one needs to check totally $2q$ functions for all top eigenvalues and eigenvectors to understand the sensitivity of a method, and vector-valued functions can be hard to interpret.

Another type of influence functions measure the robustness of PCA via the perturbation of the subspace spanned by top eigenvectors of scatter matrix estimators. In this type, instead of defining several influence functions for each eigenvalue and eigenvector, only one influence function is defined on the resulting subspace. Tanaka [1988]; Tanaka and Castaño-Tostado [1990] considered the projection matrix $\mathbf{P}$ associated with the PCA subspace. Both theoretical and sample versions of influence functions of $\mathbf{P}$ are defined for PCA, which are matrix-valued measures. This type of matrix-valued influence functions are generalized to Kernel PCA framework by Yamanishi and Tanaka [2006]. A similar definition is proposed by Bénasséni [1990]; Castaão-Tostado and Tanaka [1990]; Prendergast et al. [2008]. A real-valued influence function is defined based on the RV-coefficient (Escoufier [1973]), which is a multivariate generalization of the squared correlation coefficient, between the original subspace and the perturbed subspace.

In Section 3.2.2, we re-derive the influence measure in Prendergast et al. [2008] from the perspective of subspace distances. In Section 3.2.3, we will extend these results to Kernel PCA framework to include nonlinear dimension reduction methods. In Section 3.2.4, we will discuss the application of proposed influence measures in visualizing the sensitivity of a method, and detecting potential influential observations. In Section 3.3 we will define a sample influence function based on local rank correlations.

3.2 Empirical influence function

3.2.1 Subspaces and distance measures

We first define the subspaces and distance measures between subspaces. In $\mathbb{R}^p$, a q-dimensional linear subspace ($1 \leq q \leq p$) can be uniquely determined by an orthogonal projection matrix $\mathbf{P}$ with rank q. We can denote a subspace by

$$\mathcal{S}_{\mathbf{P}} = \left\{ \mathbf{P}\,\mathbf{x} \,:\, \mathbf{x} \in \mathbb{R}^p \right\} .$$

All q-dimensional linear subspaces in $\mathbb{R}^p$ form a Grassmannian manifold (Milnor and Stasheff [1974]),

$$\mathrm{Gr}(q, \mathbb{R}^p) = \left\{ \mathcal{S}_{\mathbf{P}} \subset \mathbb{R}^p \,:\, \mathrm{rank}(\mathbf{P}) = q \right\} .$$

A metric on this Grassmannian manifold is provided by Crone and Crosby [1995] as follows.

Definition 3.1. *Distance between subspaces with the same dimension*: Suppose $\mathcal{S}_{\mathbf{P}_1}$, $\mathcal{S}_{\mathbf{P}_2} \in \mathrm{Gr}(q, \mathbb{R}^p)$, the distance between $\mathcal{S}_{\mathbf{P}_1}$ and $\mathcal{S}_{\mathbf{P}_2}$ is define by

$$\begin{aligned} D\left(\mathcal{S}_{\mathbf{P}_1}, \mathcal{S}_{\mathbf{P}_2}\right) &= \frac{1}{\sqrt{2}} \left\|\mathbf{P}_1 - \mathbf{P}_2\right\|_F & (3.1) \\ &= \left[q - \mathrm{trace}(\mathbf{P}_1 \mathbf{P}_2)\right]^{1/2} . & (3.2) \end{aligned}$$

It has been shown in Crone and Crosby [1995] that this distance measure obeys the triangle inequality, and $0 \leq D^2\left(\mathcal{S}_{\mathbf{P}_1}, \mathcal{S}_{\mathbf{P}_2}\right) \leq \min\{q,\, p-q\}$.

3.2.2 EIF for PCA

Recall that any variant of PCA seeks a linear subspace formed by a set of orthogonal vectors. The task of sensitivity analysis in PCA is essentially to evaluate the change of resulting subspace due to the small contamination to the underlying distribution, or input data. To measure such changes, we can use the subspace distance provided in Definition 3.1 as a tool. The following notation will be useful in later discussions.

Notation

- $\mathcal{F}$: A cumulative distribution function defined on $\mathbb{R}^p$.
- $\{\mathbf{x}_i \sim \mathcal{F},\, i = 1, \ldots, n\}$: Observed data, together denoted by $\mathbf{X}_n = [\mathbf{x}_1 \cdots \mathbf{x}_n]'$.
- $\boldsymbol{\mu}$, $\boldsymbol{\Sigma}$: Expected value and covariance matrix of $\mathcal{F}$,

$$\begin{aligned} \boldsymbol{\mu}(\mathcal{F}) &= \int \mathbf{x} d\mathcal{F}(\mathbf{x}) \\ \boldsymbol{\Sigma}(\mathcal{F}) &= \int \left[\mathbf{x} - \boldsymbol{\mu}(\mathcal{F})\right]\left[\mathbf{x} - \boldsymbol{\mu}(\mathcal{F})\right]' d\mathcal{F}(x) \end{aligned}$$

- $\lambda_i(\mathcal{F})$, $\mathbf{e}_i(\mathcal{F})$: The i-th largest eigenvalue of $\boldsymbol{\Sigma}(\mathcal{F})$ and corresponding eigenvector.

- $\mathbf{U}(\mathcal{F}) = [\mathbf{e}_1\, \mathbf{e}_2\, \cdots\, \mathbf{e}_q]$: $p \times q$ matrix whose i-th column is $\mathbf{e}_i(\mathcal{F})$.
- $\mathbf{P}(\mathcal{F}) = \mathbf{U}(\mathcal{F}) \cdot \mathbf{U}(\mathcal{F})'$: Projection matrix onto the linear subspace spanned by orthogonal columns of $\mathbf{U}(\mathcal{F})$.
- $\mathcal{S}_{\mathbf{P}}$: The linear subspace characterized by the projection matrix $\mathbf{P}$.
- $\delta_{\boldsymbol{\omega}}$: The distribution giving point mass to $\boldsymbol{\omega} \in \mathbb{R}^p$.
- $\mathbf{X_{n,\boldsymbol{\omega}}}$: The contaminated sample $\mathbf{X_{n,\boldsymbol{\omega}}} = [\mathbf{x}_1, \ldots, \mathbf{x}_n, \boldsymbol{\omega}]'$.

Define PCA procedure as a statistical functional

$$T_{PCA} : \mathcal{F} \longmapsto \mathcal{S}_{\mathbf{P}(\mathcal{F})}\,,$$

which maps a multivariate distribution into a linear subspace. Unless specified otherwise, in the rest of the chapter we assume $\mathcal{F}$ is a p-dimensional distribution and $\mathcal{S}_{\mathbf{P}(\mathcal{F})}$ is a q-dimensional subspace, where q is known. Consider the contaminated model

$$\mathcal{F}_{\epsilon,\boldsymbol{\omega}} = (1-\epsilon) \cdot \mathcal{F} + \epsilon \cdot \delta_{\boldsymbol{\omega}}\,.$$

An influence function of T_{PCA} measures the changes of subspaces caused by the contamination, and it can be defined as

$$\begin{aligned} \mathrm{IF}(\boldsymbol{\omega};\, T_{PCA},\, \mathcal{F}) &= \lim_{\epsilon \to 0} \frac{D\left(T_{PCA}(\mathcal{F}),\, T_{PCA}(\mathcal{F}_{\epsilon,\boldsymbol{\omega}})\right)}{\epsilon} \\ &= \lim_{\epsilon \to 0} \frac{\left[q - \mathrm{trace}(\mathbf{P}(\mathcal{F}) \cdot \mathbf{P}(\mathcal{F}_{\epsilon,\boldsymbol{\omega}}))\right]^{1/2}}{\epsilon}. \end{aligned}$$

The explicit form of this influence function is provided in the following lemma. It is similar to the result in Bénasséni [1990] and Prendergast et al. [2008], but we will derive it from a different perspective.

Lemma 3.1. Let $\mathcal{F}$ be a cumulative distribution function defined on $\mathbb{R}^p$. Assuming $\boldsymbol{\mu}(\mathcal{F})$ and $\boldsymbol{\Sigma}(\mathcal{F})$ exist, and $\boldsymbol{\Sigma}(\mathcal{F})$ has distinct eigenvalues $\lambda_1 > \cdots > \lambda_p$, and associated eigenvectors $\mathbf{e}_1, \ldots, \mathbf{e}_p$. The influence function $\mathrm{IF}(\boldsymbol{\omega};\, T_{PCA},\, \mathcal{F})$ is given by

$$\mathrm{IF}(\boldsymbol{\omega};\, T_{PCA},\, \mathcal{F}) = \left\{ \sum_{i=1}^{q} \sum_{k=q+1}^{p} \left(\frac{a_i \cdot a_k}{\lambda_i - \lambda_k} \right)^2 \right\}^{1/2}, \tag{3.3}$$

where $a_j = \mathbf{e}_j' \cdot (\boldsymbol{\omega} - \boldsymbol{\mu})$.

Proof. Under the contaminated distribution

$$\mathcal{F}_{\epsilon,\boldsymbol{\omega}} = (1-\epsilon) \cdot \mathcal{F} + \epsilon \cdot \delta_{\boldsymbol{\omega}}\,,$$

the covariance matrix $\boldsymbol{\Sigma}(\mathcal{F}_{\epsilon,\boldsymbol{\omega}})$ can be expressed as

$$\boldsymbol{\Sigma}(\mathcal{F}_{\epsilon,\boldsymbol{\omega}}) = \boldsymbol{\Sigma}(\mathcal{F}) + \epsilon \cdot \left\{ [\boldsymbol{\omega} - \boldsymbol{\mu}(\mathcal{F})]\, [\boldsymbol{\omega} - \boldsymbol{\mu}(\mathcal{F})]' - \boldsymbol{\Sigma}(\mathcal{F}) \right\} - \epsilon^2 \cdot \left\{ [\boldsymbol{\omega} - \boldsymbol{\mu}(\mathcal{F})]\, [\boldsymbol{\omega} - \boldsymbol{\mu}(\mathcal{F})]' \right\}.$$

Using the result in Sibson [1979] and Critchley [1985], we can write the j-th eigenvector of $\boldsymbol{\Sigma}(\mathcal{F}_{\epsilon,\boldsymbol{\omega}})$ as

$$\mathbf{e}_{j,(\epsilon,\boldsymbol{\omega})} = \mathbf{e}_j - \epsilon \cdot \boldsymbol{\beta}_j + \frac{1}{2}\epsilon^2 \cdot \boldsymbol{\gamma}_j + O(\epsilon^3),$$

where

$$\begin{aligned}
\boldsymbol{\beta}_j &= -a_j \cdot \sum_{k \neq j} \left\{ \frac{a_k}{\lambda_k - \lambda_j} \cdot \mathbf{e}_k \right\}, \\
a_j &= \mathbf{e}_j' \cdot (\boldsymbol{\omega} - \boldsymbol{\mu}), \\
\boldsymbol{\gamma}_j &= -a_j^2 \cdot \left\{ \sum_{k \neq j} \frac{a_k^2}{(\lambda_k - \lambda_j)^2} \right\} \cdot \mathbf{e}_j - 2 \left\{ \sum_{k \neq j} \frac{a_k^2}{\lambda_k - \lambda_j} \right\} \cdot \boldsymbol{\beta}_j - 2a_j^3 \cdot \sum_{k \neq j} \left\{ \frac{a_k}{(\lambda_k - \lambda_j)^2} \mathbf{e}_k \right\}.
\end{aligned}$$

The distance between original subspace and perturbed subspace is calculated by

$$\begin{aligned}
D^2\left(T_{PCA}(\mathcal{F}),\, T_{PCA}(\mathcal{F}_{\epsilon,\boldsymbol{\omega}})\right) &= q - \mathrm{trace}(\mathbf{P}(\mathcal{F}) \cdot \mathbf{P}(\mathcal{F}_{\epsilon,\boldsymbol{\omega}})) \\
&= q - \sum_{i=1}^{q} \sum_{j=1}^{q} \left(\mathbf{e}_i' \cdot \mathbf{e}_{j,(\epsilon,\boldsymbol{\omega})} \right)^2,
\end{aligned}$$

where

$$\mathbf{e}_{_i}' \cdot \mathbf{e}_{_{j,(\epsilon,\boldsymbol{\omega})}} = \begin{cases} 1 + \frac{1}{2}\epsilon^2 \cdot \mathbf{e}_{_j}' \cdot \boldsymbol{\gamma}_{_j} + O(\epsilon^3), & i = j \\ -\epsilon \cdot \mathbf{e}_{_i}' \cdot \boldsymbol{\beta}_{_j}, & i \neq j \end{cases}$$

$$= \begin{cases} 1 - \frac{1}{2}\epsilon^2 \cdot \sum\limits_{k \neq j} \frac{(a_{_j} \cdot a_{_k})^2}{(\lambda_k - \lambda_j)^2} + O(\epsilon^3), & i = j \\ -\epsilon \cdot \frac{a_{_i} \cdot a_{_j}}{\lambda_i - \lambda_j} + O(\epsilon^2), & i \neq j. \end{cases}$$

Using this relation, we have

$$\sum_{j=1}^{q} \left(\mathbf{e}_{_i}' \cdot \mathbf{e}_{_{j,(\epsilon,\boldsymbol{\omega})}} \right)^2 = 1 - \epsilon^2 \cdot \sum_{k \neq i} \left(\frac{a_{_i} \cdot a_{_k}}{\lambda_i - \lambda_k} \right)^2 + \epsilon^2 \cdot \sum_{\substack{j=1 \\ j \neq i}}^{q} \left(\frac{a_{_i} \cdot a_{_j}}{\lambda_i - \lambda_j} \right)^2 + O(\epsilon^3)$$

$$= 1 - \epsilon^2 \cdot \sum_{k=q+1}^{p} \left(\frac{a_{_i} \cdot a_{_k}}{\lambda_i - \lambda_k} \right)^2 + O(\epsilon^3),$$

and

$$\sum_{i=1}^{q} \sum_{j=1}^{q} \left(\mathbf{e}_{_i}' \cdot \mathbf{e}_{_{j,(\epsilon,\boldsymbol{\omega})}} \right)^2 = q - \epsilon^2 \cdot \sum_{i=1}^{q} \sum_{k=q+1}^{p} \left(\frac{a_{_i} \cdot a_{_k}}{\lambda_i - \lambda_k} \right)^2 + O(\epsilon^3).$$

This implies that

$$D\left(\mathcal{S}_{\mathbf{P}(\mathcal{F})}, \mathcal{S}_{\mathbf{P}(\mathcal{F}_{\epsilon}, \boldsymbol{\omega})} \right) = \left\{ \epsilon^2 \cdot \sum_{i=1}^{q} \sum_{k=q+1}^{p} \left(\frac{a_{_i} \cdot a_{_k}}{\lambda_i - \lambda_k} \right)^2 + O(\epsilon^3) \right\}^{1/2},$$

and subsequently

$$\mathrm{IF}(\boldsymbol{\omega};\, T_{_{PCA}},\, \mathcal{F}) = \left\{ \sum_{i=1}^{q} \sum_{k=q+1}^{p} \left(\frac{a_{_i} \cdot a_{_k}}{\lambda_i - \lambda_k} \right)^2 \right\}^{1/2}.$$

□

From what we derived above an empirical influence function can then be obtained by replacing $\mathcal{F}$ with the empirical distribution $\widehat{\mathcal{F}}$ of observed data. Similarly, we can also obtained an empirical influence function with a certain case deleted from the sample. The results are given in the following two corollaries.

Corollary 3.1. Let $\widehat{\mathcal{F}}$ be the empirical distribution of observed data $\mathbf{X}_n$. Assuming observed data are centered, i.e. $\widehat{\boldsymbol{\mu}} = 0$, the empirical influence function of T_{PCA} is

$$\mathrm{EIF}(\boldsymbol{\omega};\, T_{PCA},\, \widehat{\mathcal{F}}) = \left\{ \sum_{i=1}^{q} \sum_{k=q+1}^{p} \left(\frac{\widehat{a}_i \cdot \widehat{a}_k}{\widehat{\lambda}_i - \widehat{\lambda}_k} \right)^2 \right\}^{1/2}, \tag{3.4}$$

where $\widehat{a}_i = \widehat{\mathbf{e}}'_i \cdot \boldsymbol{\omega}$, $\widehat{\lambda}_i$ and $\widehat{\mathbf{e}}_i$ are associated eigenvalues and eigenvectors of the sample covariance matrix $\widehat{\boldsymbol{\Sigma}}$.

Corollary 3.2. Let $\widehat{\mathcal{F}}$ be the empirical distribution of observed data $\mathbf{X}_n$. Assuming observed data are centered, i.e. $\widehat{\boldsymbol{\mu}} = 0$, the empirical influence function of T_{PCA} with the j-th case deleted is

$$\mathrm{EIF}_{(j)}(T_{PCA},\, \widehat{\mathcal{F}}) = \left\{ \sum_{i=1}^{q} \sum_{k=q+1}^{p} \left(\frac{\widehat{\mathbf{e}}'_i\, \mathbf{x}_j\, \mathbf{x}'_j \widehat{e}_k}{\widehat{\lambda}_i - \widehat{\lambda}_k} \right)^2 \right\}^{1/2}, \tag{3.5}$$

where $\widehat{\lambda}_i$ and $\widehat{\mathbf{e}}_i$ are associated eigenvalues and eigenvectors of $\widehat{\boldsymbol{\Sigma}}$.

As can be easily seen from equation (3.1) and (3.3), both theoretical and empirical influence measures of PCA are unbounded, indicating the non-robustness of the traditional PCA. Influence measures can be generally defined on robust PCA variants based on some robust scatter matrix estimators.

Consider a PCA variant which is based on the eigendecomposition of an affine equivariant scatter matrix functional $\mathbf{C}(\mathcal{F})$. The resulting subspace is spanned by the top q eigenvectors of $\mathbf{C}(\mathcal{F})$, and we denote this subspace by $\mathcal{S}_{\mathbf{C}}(\mathcal{F})$. Define the functional

$$T_C : \mathcal{F} \mapsto \mathcal{S}_{\mathbf{C}}(\mathcal{F})\,.$$

Assume that under the contamination model $\mathcal{F}_{\epsilon,\boldsymbol{\omega}}$, the functional $\mathbf{C}$ has a perturbation of form

$$\mathbf{C}(\mathcal{F}_{\epsilon,\boldsymbol{\omega}}) = \mathbf{C}(\mathcal{F}) + \epsilon \cdot \mathbf{C}_1 + O(\epsilon^2)\,.$$

Using the result of perturbation expansions of the projector matrix in Tanaka and Castaño-Tostado [1990], it can be shown that the influence function of $T_{_C}$ has an explicit form

$$\mathrm{IF}(\boldsymbol{\omega};\, T_{_C},\, \mathcal{F}) = \left\{ \sum_{i=1}^{q} \sum_{j=q+1}^{p} \left(\frac{\mathbf{v}_{_i}' \cdot \mathbf{C}_1 \cdot \mathbf{v}_{_j}}{\kappa_{_i} - \kappa_{_j}} \right)^2 \right\}^{1/2} , \tag{3.6}$$

where $\kappa_{_i}$ and $\mathbf{v}_{_i}$ are associated eigenvalues and eigenvectors of $\mathbf{C}(\mathcal{F})$. A similar result is provided in Prendergast et al. [2008], which was derived from RV-coefficient.

3.2.3 EIF for Kernel PCA

As reviewed in Section 1.2.5, Kernel PCA provides a unified framework of dimension reduction. The sensitivity analysis of nonlinear dimension reduction methods can be performed under this framework.

Kernel PCA assumes a feature map

$$\Phi : \mathbb{R}^p \to \mathcal{H}\, ,$$

which transforms the observed data into a Hilbert space whose dimension t can be arbitrarily large, and possibly infinite. The transformed data $\{\Phi(\mathbf{x}_{_i})_{t\times 1} \in \mathcal{H};\, i = 1, \ldots, n\}$ together are denoted by a matrix $\Phi(\mathbf{X})_{n\times t} = [\Phi(\mathbf{x}_{_1}) \cdots \Phi(\mathbf{x}_{_n})]'$. The kernel function

$$k(\cdot, \cdot) : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$$

defines the inner product in $\mathcal{H}$, i.e. $k(\mathbf{x}, \mathbf{y}) = \langle \Phi(\mathbf{x}), \Phi(\mathbf{y}) \rangle$. The $n \times n$ matrix $\mathbf{K}$, whose (i, j) element is $K_{ij} = k(\mathbf{x}_{_i}, \mathbf{x}_{_j})$, is called the kernel matrix. Traditional PCA is then performed in the feature space via the eigendecomposition of the kernel matrix to obtain the low-dimensional representation.

Assuming the transformed data are centered, i.e. $\sum_{i=1}^{n} \Phi(\mathbf{x}_{_i}) = 0$, the resulting subspace in $\mathcal{H}$ is spanned by the top q eigenvectors $\left\{\widehat{\mathbf{v}}_{_1}, \cdots, \widehat{\mathbf{v}}_{_q}\right\}$ of the covariance operator

$$\mathbf{C}_\Phi = \frac{1}{n} \sum_{i=1}^{n} \Phi(\mathbf{x}_{_i}) \cdot \Phi(\mathbf{x}_{_i})' \,.$$

Define Kernel PCA as a statistical functional

$$\widehat{T}_{_{KPCA}} : \widehat{\mathcal{F}} \rightarrow \text{span}\left\{\widehat{\mathbf{v}}_{_1}, \cdots, \widehat{\mathbf{v}}_{_q}\right\} \,,$$

and the result in Corollary 3.1 can be directly generalized to obtain the empirical influence function of Kernel PCA, i.e.

$$\text{EIF}(\boldsymbol{\omega}; \widehat{T}_{_{KPCA}}, \widehat{\mathcal{F}}) = \left\{ \sum_{i=1}^{q} \sum_{j=q+1}^{t} \left(\frac{\widehat{a}_{_i} \cdot \widehat{a}_{_j}}{\widehat{\lambda}_i - \widehat{\lambda}_j} \right)^2 \right\}^{1/2} \,, \tag{3.7}$$

where $\widehat{a}_{_j} = \widehat{\mathbf{v}}'_{_j} \cdot \Phi(\boldsymbol{\omega})$, $\widehat{\lambda}_j$ and $\widehat{\mathbf{v}}_{_j}$ are associated eigenvalues and eigenvectors of $\mathbf{C}_\Phi$. However, this result cannot be directly used because in practice the feature map Φ is usually implicit and therefore $\mathbf{C}_\Phi$, $\widehat{\lambda}_j$ and $\widehat{\mathbf{v}}_{_j}$ are also implicit. Recall equation (1.13) and (1.14), for any positive eigenvalue $\widehat{\lambda}_j > 0$, the corresponding eigenvector $\widehat{\mathbf{v}}_{_j}$ lies in the span of $\{\Phi(\mathbf{x}_{_1}), \ldots, \Phi(\mathbf{x}_{_n})\}$, which allows us to rewrite $\widehat{\lambda}_j$ and $\widehat{\mathbf{v}}_{_j}$ as

$$\begin{aligned} \widehat{\mathbf{v}}_{_j} &= c_{_j} \cdot \Phi(\mathbf{X})' \widehat{\boldsymbol{\alpha}}_j \,, \\ n\,\widehat{\lambda}_j &= \widehat{\xi}_j \,, \end{aligned}$$

where $\widehat{\xi}_j$ and $\widehat{\boldsymbol{\alpha}}_j = [\widehat{\alpha}_{j1}, \cdots, \widehat{\alpha}_{jn}]'$ are the j-th eigenvalue and eigenvector of the kernel matrix $\mathbf{K}$ respectively, and $c_{_j}$ is the normalizing factor. Therefore we have

$$\begin{aligned} \widehat{a}_{_j} &= \widehat{\mathbf{v}}'_{_j} \cdot \Phi(\boldsymbol{\omega}) \\ &= c_{_j} \cdot \underset{1\times n}{\widehat{\boldsymbol{\alpha}}'_j} \, \underset{n\times t}{\Phi(\mathbf{X})} \, \underset{t\times 1}{\Phi(\boldsymbol{\omega})} \\ &= c_{_j} \cdot \sum_{k=1}^{n} \widehat{\alpha}_{jk} \cdot k(\mathbf{x}_{_k}, \boldsymbol{\omega}) \,. \end{aligned} \tag{3.8}$$

The normalizing factor $c_{_j}$ is obtained by

$$\begin{aligned}
&\widehat{\mathbf{v}}'_{_j} \cdot \widehat{\mathbf{v}}_{_j} = 1 \\
\Rightarrow &c_{_j}^2 \cdot \widehat{\boldsymbol{\alpha}}'_j \, \Phi(\mathbf{X}) \, \Phi(\mathbf{X})' \, \widehat{\boldsymbol{\alpha}}_j = 1 \\
\Rightarrow &c_{_j}^2 \cdot \widehat{\boldsymbol{\alpha}}'_j \, \mathbf{K} \, \widehat{\boldsymbol{\alpha}}_j = 1 \\
\Rightarrow &c_{_j}^2 \cdot \widehat{\xi}_j = 1 \\
\Rightarrow &c_{_j} = 1/\sqrt{\widehat{\xi}_j} \, . \qquad (3.9)
\end{aligned}$$

Substituting equation (3.8) and (3.9) into (3.7), the respective empirical influence function can be written as

$$\mathrm{EIF}(\boldsymbol{\omega};\, \widehat{T}_{_{KPCA}},\, \widehat{\mathfrak{F}}) = \left\{ \sum_{i=1}^{q} \left[\sum_{j=q+1}^{n} \left(\frac{n \cdot \widehat{a}_{_i} \cdot \widehat{a}_{_j}}{\widehat{\xi}_i - \widehat{\xi}_j} \right)^2 + \underbrace{\sum_{s=n+1}^{t} \left(\frac{\widehat{a}_{_i} \cdot \widehat{a}_{_s}}{\widehat{\lambda}_i - \widehat{\lambda}_s} \right)^2}_{(*)} \right] \right\}^{1/2} , \qquad (3.10)$$

where

$$\widehat{a}_{_j} = \begin{cases} \sum_{k=1}^{n} \widehat{\alpha}_{jk} \cdot k(\mathbf{x}_{_k}, \boldsymbol{\omega}) / \sqrt{\widehat{\xi}_j} & j \leq n \\ \widehat{\mathbf{v}}'_{_j} \cdot \Phi(\boldsymbol{\omega}) & j > n. \end{cases}$$

Note that only top n eigenvalues of $\mathbf{C}_\Phi$ are positive, and all remaining eigenvalues are 0. Equation (3.10) can still not be used because the kernel trick can only be applied on eigenvectors corresponding to positive eigenvalues, and $\widehat{a}_{_j}$ for any $j > n+1$ is still implicit. Therefore, part $(*)$ in equation (3.10) is not available. If we assume zero eigenvalues and corresponding eigenvectors carry no information about the resulting subspace, then we can truncate the summation, and approximate the empirical influence function in equation (3.10) by only top n terms, i.e., it is redefined as

$$\mathrm{EIF}(\boldsymbol{\omega};\, \widehat{T}_{_{KPCA}},\, \widehat{\mathfrak{F}}) = \left\{ \sum_{i=1}^{q} \sum_{j=q+1}^{n} \left(\frac{n \cdot \widehat{a}_{_i} \cdot \widehat{a}_{_j}}{\widehat{\xi}_i - \widehat{\xi}_j} \right)^2 \right\}^{1/2} , \qquad (3.11)$$

where

$$\widehat{a}_{j} = \frac{1}{\sqrt{\widehat{\xi}_j}} \sum_{k=1}^{n} \widehat{\alpha}_{jk} \cdot k(\mathbf{x}_{k}, \boldsymbol{\omega}).$$

Equation (3.11) suggests that if an unbounded kernel $k(\cdot,\cdot)$ is used, the Kernel PCA would have an unbounded empirical influence function, which implies the non-robustness of the method.

As reviewed in Section 1.2.5, many nonlinear dimension reduction methods can be described as special cases under the Kernel PCA framework, where these methods only specify some kernel matrix $\mathbf{K}$ without an explicit kernel $k(\cdot,\cdot)$. For example, given n data points, ISOMAP first constructs an approximated squared geodesic distance matrix $\mathbf{D}^{\mathcal{G}}$ from the Dijkstra's algorithm, and then obtains the kernel matrix $\mathbf{K}_n^{ISO}$ as

$$\mathbf{K}_n^{ISO} = -\frac{1}{2}(\mathbf{I}_n - \mathbf{e}_{n}\mathbf{e}'_{n})\mathbf{D}_n^{\mathcal{G}}(\mathbf{I}_n - \mathbf{e}_{n}\mathbf{e}'_{n}),$$

where $\mathbf{e}_{n} = n^{-1/2}(1,\dots,1)'$ and $\mathbf{I}_n$ is the identity matrix. In this case, we can rebuild the kernel matrix $\mathbf{K}_{n+1,\boldsymbol{\omega}}^{ISO}$ based on the contaminated sample $\{\mathbf{x}_{1},\dots,\mathbf{x}_{n},\boldsymbol{\omega}\}$, and the terms $k(\mathbf{x}_{k},\boldsymbol{\omega})$ in equation (3.11) can be approximated by the last column of $\mathbf{K}_{n+1,\boldsymbol{\omega}}^{ISO}$.

Also note that in Kernel PCA framework, theoretically the kernel matrix $\mathbf{K}$ is required to be positive semidefinite. However in some methods, for example ISOMAP, which do not specify a kernel $k(\cdot,\cdot)$, there is no guarantee that $\mathbf{K}^{ISO}$ is positive semidefinite (Ham et al. [2004]). In other words, some eigenvalues of $\mathbf{K}^{ISO}$ could be negative. In this case, we will discard any terms that involve negative eigenvalues in equation (3.11).

Similarly, we can obtain the empirical influence function of Kernel PCA with the s-th case deleted. It is

$$\mathrm{EIF}_{(s)}(\widehat{T}_{KPCA}, \widehat{\mathfrak{F}}) = \left\{ \sum_{i=1}^{q} \sum_{j=q+1}^{n} \left(\frac{n \cdot \widehat{a}_{i} \cdot \widehat{a}_{j}}{\widehat{\xi}_i - \widehat{\xi}_j} \right)^2 \right\}^{1/2}, \tag{3.12}$$

where

$$\widehat{a}_{_j} = \frac{1}{\sqrt{\widehat{\xi}_j}} \sum_{k \neq s} \widehat{\alpha}_{jk} \cdot k(\mathbf{x}_{_k}, \mathbf{x}_{_s}).$$

3.2.4 Visualizing the influence measure and detecting influential observations

In this section, we discuss some graphical display strategies for visualizing influence measures and for flagging potential influential observations.

First we discuss how to plot the influence function of PCA. When we display the influence function $\mathrm{IF}(\boldsymbol{\omega};\, T_{_{PCA}},\, \mathcal{F})$, at each graph we can at most see $\boldsymbol{\omega}$ from a plane in $\mathbb{R}^p$. An immediate question is that for a large value of input dimension p, which plane shall we choose to display the influence function. Note that in a given direction, the influence measure $\mathrm{IF}(\boldsymbol{\omega};\, T_{_{PCA}},\, \mathcal{F})$ is proportional to the squared length of $\boldsymbol{\omega}$, i.e. if $\boldsymbol{\omega}_1 = c \cdot \boldsymbol{\omega}_2$, we have

$$\mathrm{IF}(\boldsymbol{\omega}_1;\, T_{_{PCA}},\, \mathcal{F}) = c^2 \cdot \mathrm{IF}(\boldsymbol{\omega}_2;\, T_{_{PCA}},\, \mathcal{F})\,,$$

assuming $\boldsymbol{\mu}(\mathcal{F}) = 0$. Therefore, we are interested to find the most influential direction in PCA for graphical display. The following lemma provides a guideline for plotting the influence function.

Lemma 3.2. Let $\mathcal{F}$ be a cumulative distribution function defined on $\mathbb{R}^p$. Assume $\boldsymbol{\mu}(\mathcal{F}) = 0$ and $\boldsymbol{\Sigma}(\mathcal{F})$ exist. Also assume $\boldsymbol{\Sigma}(\mathcal{F})$ has distinct eigenvalues $\lambda_1 > \cdots > \lambda_p$, and associated eigenvectors $\mathbf{e}_{1}, \ldots, \mathbf{e}_{p}$. The most influential direction $\mathbf{z}^{*}$ in PCA can be defined as

$$\begin{aligned}
\mathbf{z}^{*} &= \underset{\|\mathbf{z}\|=1}{\arg\max} \left\{ \mathrm{IF}(\mathbf{z};\, T_{PCA},\, \mathcal{F}) \right\} \\
&= \underset{\|\mathbf{z}\|=1}{\arg\max} \left\{ \sum_{i=1}^{q} \sum_{j=q+1}^{p} \left(\frac{\mathbf{e}_{i}' \cdot \mathbf{z}\,\mathbf{z}' \cdot \mathbf{e}_{j}}{\lambda_i - \lambda_j} \right)^{2} \right\} . \qquad (3.13)
\end{aligned}$$

The solution to equation (3.13) is

$$\mathbf{z}^{*} = \pm \frac{1}{\sqrt{2}} \mathbf{e}_{q} \pm \frac{1}{\sqrt{2}} \mathbf{e}_{q+1} .$$

Proof. The eigenvectors $\left\{ \mathbf{e}_{1}, \ldots, \mathbf{e}_{q} \right\}$ form an orthogonal basis of $\mathbb{R}^{p}$, therefore any unit vector $\mathbf{z}$ can be expressed as $\mathbf{z} = \sum_{i=1}^{p} a_{i} \cdot \mathbf{e}_{i}$. Let $b_{i} = a_{i}^{2}$, and $\lambda_{(i,j)} = (\lambda_i - \lambda_j)^2$, then the maximization problem in (3.13) is equivalent to

$$\begin{aligned}
&\max \left\{ \sum_{i=1}^{q} \sum_{k=q+1}^{p} \frac{b_{i} \cdot b_{j}}{\lambda_{(i,j)}} \right\} , \\
&\text{s.t. } \sum_{i=1}^{p} b_{i} = 1 , \\
&\qquad b_{i} \geq 0,\ i = 1, \ldots, p .
\end{aligned}$$

Since $\lambda_1 > \cdots > \lambda_p$, for any $1 \leq i \leq q$ and $q+1 \leq j \leq p$, we have

$$\lambda_{(q,q+1)} \leq \lambda_{(i,j)} .$$

Then we have

$$\begin{aligned}
\sum_{i=1}^{q} \sum_{k=q+1}^{p} \frac{b_{i} \cdot b_{j}}{\lambda_{(i,j)}} &\leq \frac{1}{\lambda_{(q,q+1)}} \sum_{i=1}^{q} \sum_{k=q+1}^{p} b_{i} \cdot b_{j} \\
&= \frac{1}{\lambda_{(q,q+1)}} \left(\sum_{i=1}^{q} b_{i} \right) \cdot \left(\sum_{k=q+1}^{p} b_{j} \right) \\
&\leq \frac{1}{4\, \lambda_{(q,q+1)}} .
\end{aligned}$$

The equality holds if $b_q = b_{q+1} = 1/2$ and $b_j = 0$ for any other j. Thus, the solution to equation (3.13) is

$$\mathbf{z}^* = \pm\frac{1}{\sqrt{2}}\mathbf{e}_q \pm \frac{1}{\sqrt{2}}\mathbf{e}_{q+1}\,.$$

$\square$

Lemma 3.2 suggests that adding a vector $\boldsymbol{\omega}$ of a fixed length through the direction of $\mathbf{z}^*$ will affect the PCA subspace the most. Thus, to visualize the effect of an added point to PCA or robust PCA variants, the plane spanned by $\left\{\frac{1}{\sqrt{2}}\mathbf{e}_q + \frac{1}{\sqrt{2}}\mathbf{e}_{q+1}, \frac{1}{\sqrt{2}}\mathbf{e}_q - \frac{1}{\sqrt{2}}\mathbf{e}_{q+1}\right\}$ is the most appropriate for displaying the influence measure. We illustrate our strategy via a simple example.

Example 3.1. *Influence function of PCA*: We consider a 5-dimensional multivariate normal distribution $\mathcal{F} = \mathcal{N}_5(0, \boldsymbol{\Sigma})$, where $\boldsymbol{\Sigma} = \mathrm{diag}([8, 6, 4, 0.5, 0.1])$. In this case, $q = 3$ and the true subspace is spanned by the top three eigenvectors $\mathbf{v}_1$, $\mathbf{v}_2$, $\mathbf{v}_3$, where the i-th eigenvector $\mathbf{v}_i$ of $\boldsymbol{\Sigma}$ is a $p \times 1$ vector whose i-th element equals 1 and all other elements equal 0. We display the influence function $\mathrm{IF}(\boldsymbol{\omega}; T_{PCA}, \mathcal{F})$ on four planes spanned by different combinations of eigenvectors. As can be seen from Figure 3.1, the top left panel shows the most influential directions, which coincides with the result in Lemma 3.2. On the other hand, adding a point to the plane spanned by $\mathbf{v}_2$ and $\mathbf{v}_3$ will not affect the resulting subspace and therefore have 0 influence measure.

We can also plot the empirical influence function of Kernel PCA for visualization. Consider the same distribution $\mathcal{F} = \mathcal{N}_5(0, \boldsymbol{\Sigma})$ in Example 3.1, a sample of size $n = 100$ is generated from $\mathcal{F}$. In Figure 3.2 we plot the empirical influence function of Kernel PCA with a polynomial kernel

$$k(\mathbf{x}, \mathbf{y}) = (\mathbf{x}'\,\mathbf{y} + 1)^2\,.$$

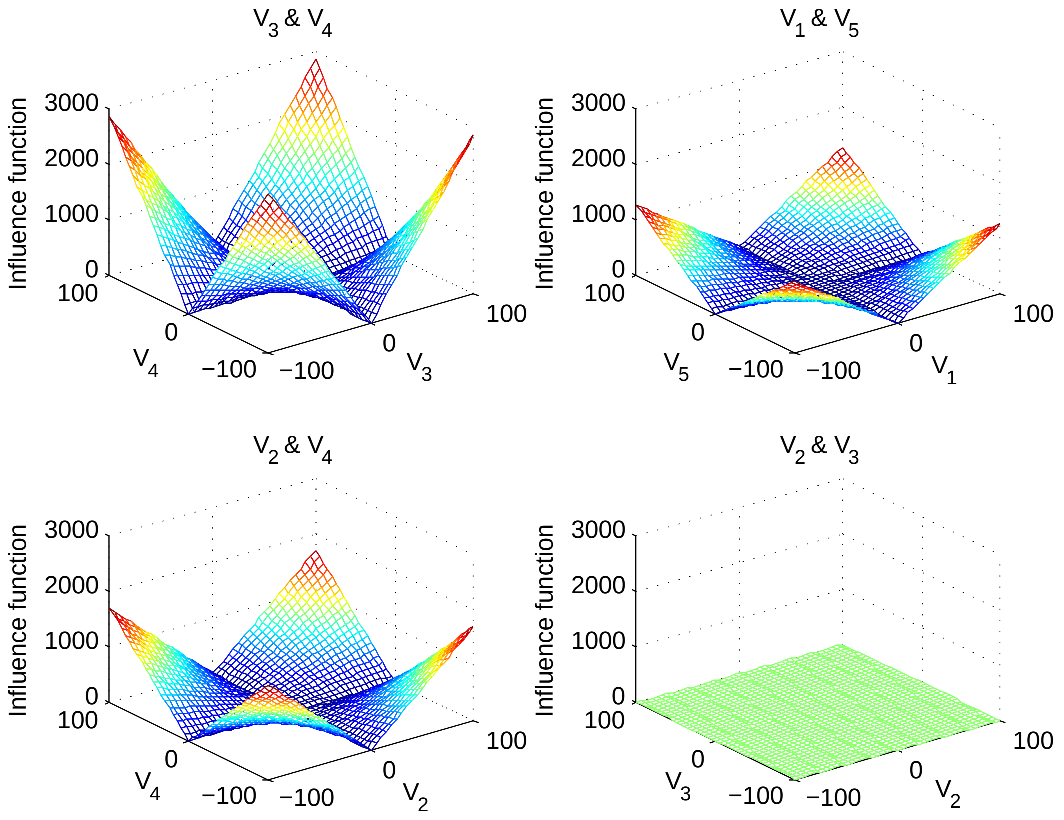

Figure 3.1: Visualizing the influence function of PCA

In Figure 3.3, we plot empirical influence functions of LLE and ISOMAP as special cases of Kernel PCA. As can be seen, both Kernel PCA with the polynomial kernel and ISOMAP have an unbounded empirical influence measure since their associated kernels are unbounded. On the other hand, LLE, whose associated kernel matrix is bounded, has a bounded empirical influence measure. It suggests that LLE is more resistant to outliers than ISOMAP and Kernel PCA with the polynomial kernel. However, the idea of finding the most influential direction for displaying the influence function is more challenging to formulate in Kernel PCA. It is much more difficult to obtain an explicit result such as

Lemma 3.2 as a guideline. A different strategy for plotting influence measures of Kernel PCA will be discussed in Section 3.3.

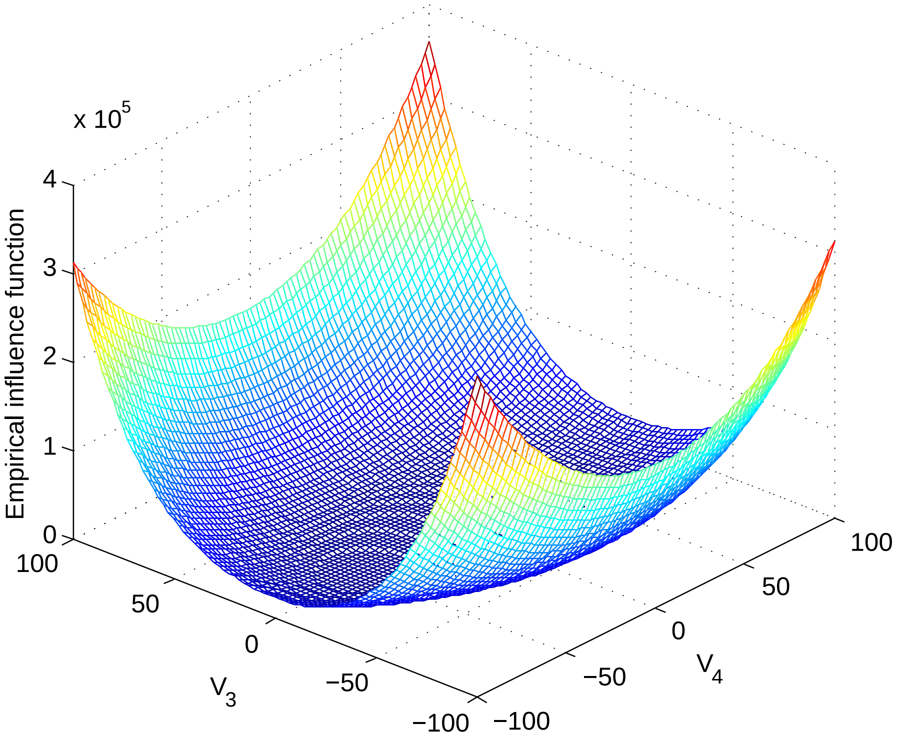

Figure 3.2: The empirical influence function of Kernel PCA with a polynomial kernel

A second task of sensitivity analysis in PCA is to flag influential observations based on the empirical influence function $\mathrm{EIF}_{(j)}(T_{PCA}, \widehat{\mathfrak{F}})$. A simple way is to plot $\mathrm{EIF}_{(j)}$ for each case j, and select those with high values of influence as candidates. However, since sample estimates of eigenvalues and eigenvectors, which are not robust against outliers, are used in $\mathrm{EIF}_{(j)}(T_{PCA}, \widehat{\mathfrak{F}})$, the sensitivity measure $\mathrm{EIF}_{(j)}$ itself might be heavily influenced by outliers. Thus, this simple strategy could fail when several outliers appear at the same

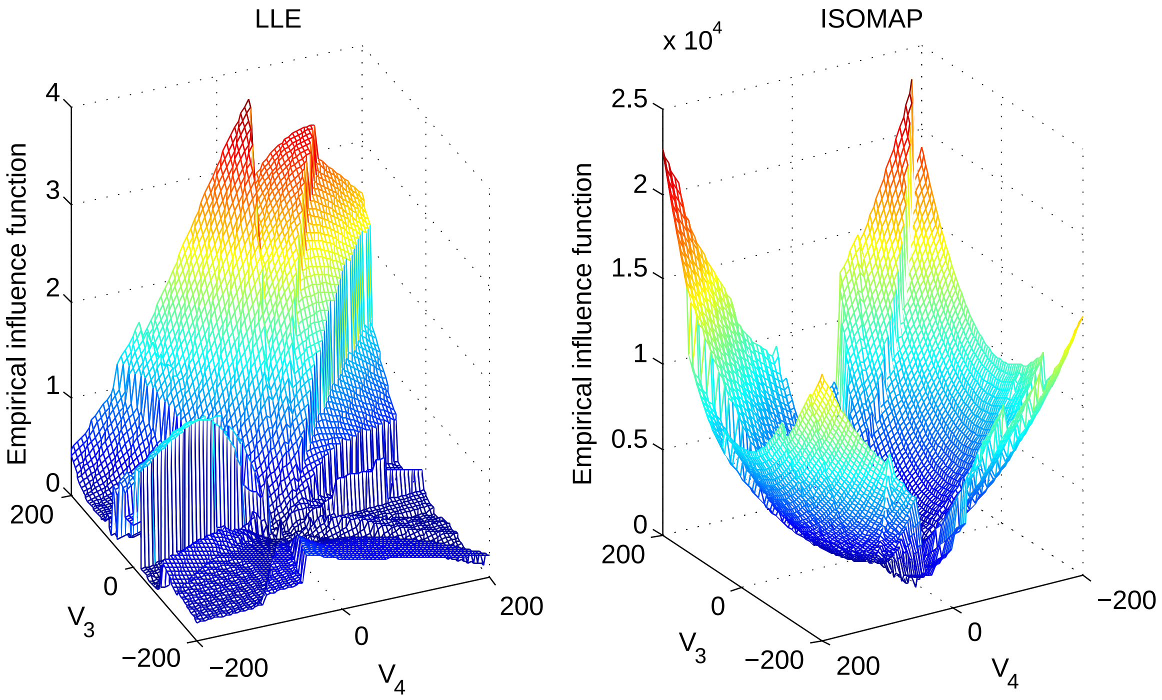

Figure 3.3: The empirical influence function of LLE and ISOMAP

time, and this can be understood as a masking effect, where multiple outliers may affect the sensitivity measure enough such that no points are declared as outliers. A more practical strategy is to replace the sample estimates of $\widehat{\lambda_i}$ and $\widehat{\mathbf{e}}_i$ in $\text{EIF}_{(j)}(T_{PCA}, \widehat{\mathfrak{F}})$ by some robust alternatives. We illustrate this argument in the following example.

Example 3.2. *Detecting influential observations*: We generate a sample of size $n = 50$ from a contaminated multivariate normal distribution,

$$\mathbf{x}_i \sim (1-\epsilon)\cdot\mathcal{N}_5(0,\mathbf{\Sigma}) + \epsilon\cdot\mathcal{N}_5(\boldsymbol{\mu}_1,\mathbf{\Sigma}_1)$$

where $\mathbf{\Sigma} = \text{diag}([8,6,4,0.5,0.1])$, $\boldsymbol{\mu}_1 = [0,0,5,5,0]'$, $\mathbf{\Sigma}_1 = 0.5\cdot\mathbf{I}$, and the contamination level $\epsilon = 0.1$. In Figure 3.4, we plot in the left panel the empirical influence function

$\mathrm{EIF}_{(j)}(T_{PCA}, \widehat{\mathcal{F}})$ defined in equation (3.5), and in the right panel we plot $\mathrm{EIF}_{(j)}(T_{PCA}, \widehat{\mathcal{F}})$ with $\widehat{\lambda_i}$ and $\hat{\mathbf{e}}_i$ replaced by true values λ_i and $\mathbf{e}_i$ from $\mathcal{N}_5(0, \boldsymbol{\Sigma})$. Outliers generated from $\mathcal{N}_5(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$ are marked by red in both bar plots. As can be seen, directly using $\mathrm{EIF}_{(j)}(T_{PCA}, \widehat{\mathcal{F}})$ will suffer from masking effect and possibly no outliers would get picked out. On the other hand, all outliers could be flagged when we cheated by using true values of eigenvalues and eigenvectors. It can be expected that if robust estimates of eigenvalues and eigenvectors are used, the result would be close to that in the right panel. Note that the same strategy can be played in Kernel PCA to detect the influential observations.

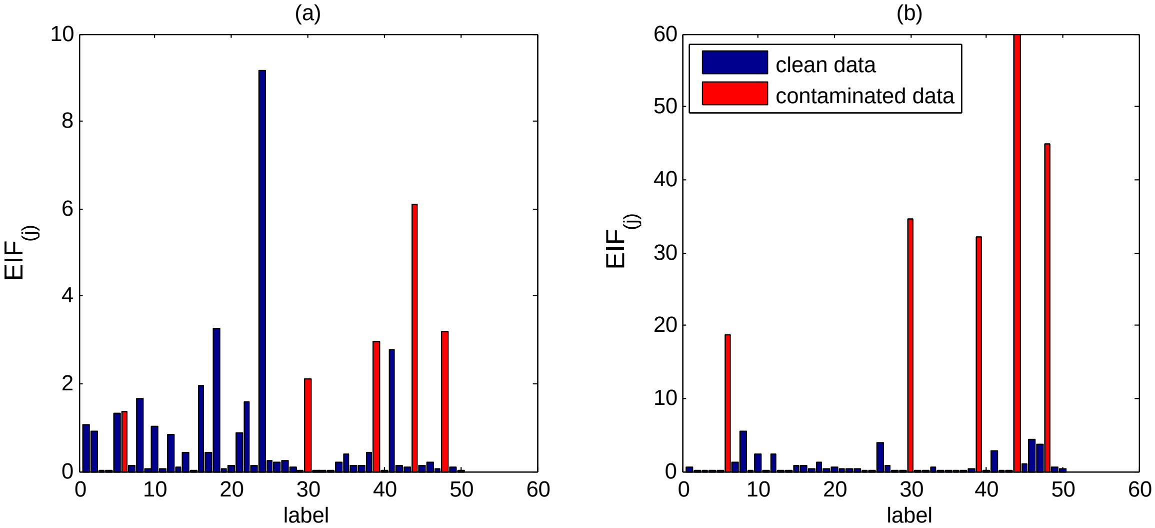

Figure 3.4: Detecting influential observations. (a) $\mathrm{EIF}_{(j)}$ with sample estimates of eigenvalues and eigenvectors. (b) $\mathrm{EIF}_{(j)}$ with true values of eigenvalues and eigenvectors.

3.3 Sample influence functions based on local rank correlations

There is an interesting phenomenon in influence functions provided in the previous section. Take equation (3.3) for example, the influence function of PCA has an explicit form

$$\mathrm{IF}(\boldsymbol{\omega};\, T_{PCA},\, \mathcal{F}) = \left\{ \sum_{i=1}^{q} \sum_{k=q+1}^{p} \left(\frac{a_i \cdot a_k}{\lambda_i - \lambda_k} \right)^2 \right\}^{1/2},$$

where $a_j = \mathbf{e}_j' \cdot (\boldsymbol{\omega} - \boldsymbol{\mu})$. Now suppose $\boldsymbol{\mu} = 0$ and we add a point $\boldsymbol{\omega}$ through the direction of the bottom eigenvector $\mathbf{e}_p$, i.e. $\boldsymbol{\omega} = c \cdot \mathbf{e}_p$. Clearly, in this case $a_p = c$ and $a_j = 0$ for any $j \neq p$. Therefore the influence function $\mathrm{IF}(c \cdot \mathbf{e}_p;\, T_{PCA},\, \mathcal{F}) = 0$, suggesting that there is no influence by adding $\boldsymbol{\omega} = c \cdot \mathbf{e}_p$, which is obviously not true. The reason is that adding $\boldsymbol{\omega} = c \cdot \mathbf{e}_p$ will not change the direction of any eigenvector, but potentially the order of them. The resulting PCA subspace spanned by top eigenvectors will not change until the length c is large enough to make the old bottom eigenvector $\mathbf{e}_p$ into top q. In other words, the change of PCA subspace due to $\boldsymbol{\omega} = c \cdot \mathbf{e}_p$ is not continuous in c. However, the influence function (or empirical influence function) is defined as the Gâteaux derivative of the PCA functional, therefore it cannot capture this discontinuous change.

Besides, empirical influence functions are not applicable for those dimension reduction methods without explicit expressions, or cannot be described as Kernel PCA, especially some robust dimension reduction methods obtained from iterative algorithms. These facts suggest that although the empirical influence function is a useful tool in the sensitivity analysis of dimension reduction methods, there is still room for other types of influence measures to be developed. In this section, we will define another type of influence measure for dimension reduction methods with broader applicability. The proposed measure is defined without any assumption of underlying distribution, and it is based on the performance measure $G_J(\psi, \mathbf{X})$ provided in Chapter 2.

The local rank correlation measures the resemblance between the original data and an output low-dimensional representation of a given method. The change of the output of the given method due to an added (or deleted) case can be quantified by the decrease (or increase) of the local rank correlation. Motivated by the definition of the sensitivity curve, we can define a finite sample version of influence measure. Since this influence measure is defined from a pure sample aspect, we name this measure "sample influence function".

Definition 3.2. *Sample influence function*: For a given dataset $\mathbf{X}_n$ in $\mathbb{R}^p$ of size n, denoted by

$$\mathbf{X}_{n,\boldsymbol{\omega}} = [\mathbf{x}_1, \ldots, \mathbf{x}_n, \boldsymbol{\omega}]',$$

the original data adjoined to a new data point $\boldsymbol{\omega} \in \mathbb{R}^p$. Also denote by

$$\mathbf{X}_{(-i)} = [\mathbf{x}_1, \ldots, \mathbf{x}_{i-1}, \mathbf{x}_{i+1}, \ldots, \mathbf{x}_n]' .$$

For a given dimensionality reduction method $\widehat{\psi} : \mathbb{R}^p \to \mathbb{R}^q$, the sample influence function of $\widehat{\psi}$ with adding $\boldsymbol{\omega}$ and the sample influence function of $\widehat{\psi}$ with deleting $\mathbf{x}_i$ are defined as

$$\mathrm{SIF}(\boldsymbol{\omega};\, \widehat{\psi},\, \mathbf{X}_n) = G_J(\widehat{\psi}, \mathbf{X}_{n,\boldsymbol{\omega}}) - G_J(\widehat{\psi}, \mathbf{X}_n)\,, \tag{3.14}$$

$$\mathrm{SIF}_{(i)}(\widehat{\psi},\, \mathbf{X}_n) = G_J(\widehat{\psi}, \mathbf{X}_{(-i)}) - G_J(\widehat{\psi}, \mathbf{X}_n). \tag{3.15}$$

where G_J can be any local rank correlation defined in Definition 2.1 and 2.2.

Unlike the empirical influence function, the sample influence function $\mathrm{SIF}(\boldsymbol{\omega};\, \widehat{\psi},\, \mathbf{X}_n)$ directly measures the change of the quality of a given method caused by an added (or deleted) point. It takes values between -1 and 1, where $\mathrm{SIF}(\boldsymbol{\omega};\, \widehat{\psi},\, \mathbf{X}_n) = 0$ indicates that the point $\boldsymbol{\omega}$ is not influential to the embedding. The positive value of $\mathrm{SIF}(\boldsymbol{\omega};\, \widehat{\psi},\, \mathbf{X}_n)$ indicates that the performance of $\widehat{\psi}$ is improved by adding $\boldsymbol{\omega}$, and the negative value indicates the decrease. Note a key difference between the use of the empirical influence function and the sample influence function is that, the empirical influence function of a robust dimension reduction method should be bounded, while according to the sample influence function, a robust method $\widehat{\psi}$ is desired to have influence measure $\mathrm{SIF}(\boldsymbol{\omega};\, \widehat{\psi},\, \mathbf{X}_n)$ uniformly close to 0.

We will illustrate in the following example that how one can use the sample influence function to compare the robustness of different dimensionality reduction methods.

Example 3.3. *Sample influence function of PCA, ROBPCA, LLE, and ISOMAP*: Consider the 5-dimensional multivariate normal distribution $\mathcal{F} = \mathcal{N}_5(0, \boldsymbol{\Sigma})$ in Example 3.1. We generate a sample of size $n = 50$ from $\mathcal{F}$, and apply four different methods, PCA, ROBPCA (Hubert et al. [2005]), LLE and ISOMAP. The outlier $\boldsymbol{\omega}$ is added on the plane spanned by the third and fourth eigenvector $\mathbf{v}_3$ and $\mathbf{v}_4$ of $\boldsymbol{\Sigma}(\mathcal{F})$. The sample influence functions $\mathrm{SIF}(\boldsymbol{\omega};\, \widehat{\psi},\, \mathbf{X}_n)$ are calculated, respectively, and plotted over $[-1000, 1000] \times [-1000, 1000]$ in Figure 3.5.

As can be seen from Figure 3.5, among these four methods, PCA is the one that affected by the outlier most easily. We can observe an obvious decrease in the sample influence function of PCA when the outlier appears through the direction of $\mathbf{v}_4$. ISOMAP is also influenced by the outlier, but is slightly more stable than PCA. On the other hand, ROBPCA and LLE are much more insensitive to the outlier. The change of $\mathrm{SIF}(\boldsymbol{\omega};\, \widehat{\psi}_{LLE},\, \mathbf{X}_n)$ is smaller compared to PCA and ISOMAP, and the performance of ROBPCA barely drop as indicated by the sample influence functions. Note that the resulting subspace in ROBPCA does not have an explicit expression, and thus the empirical influence function of ROBPCA

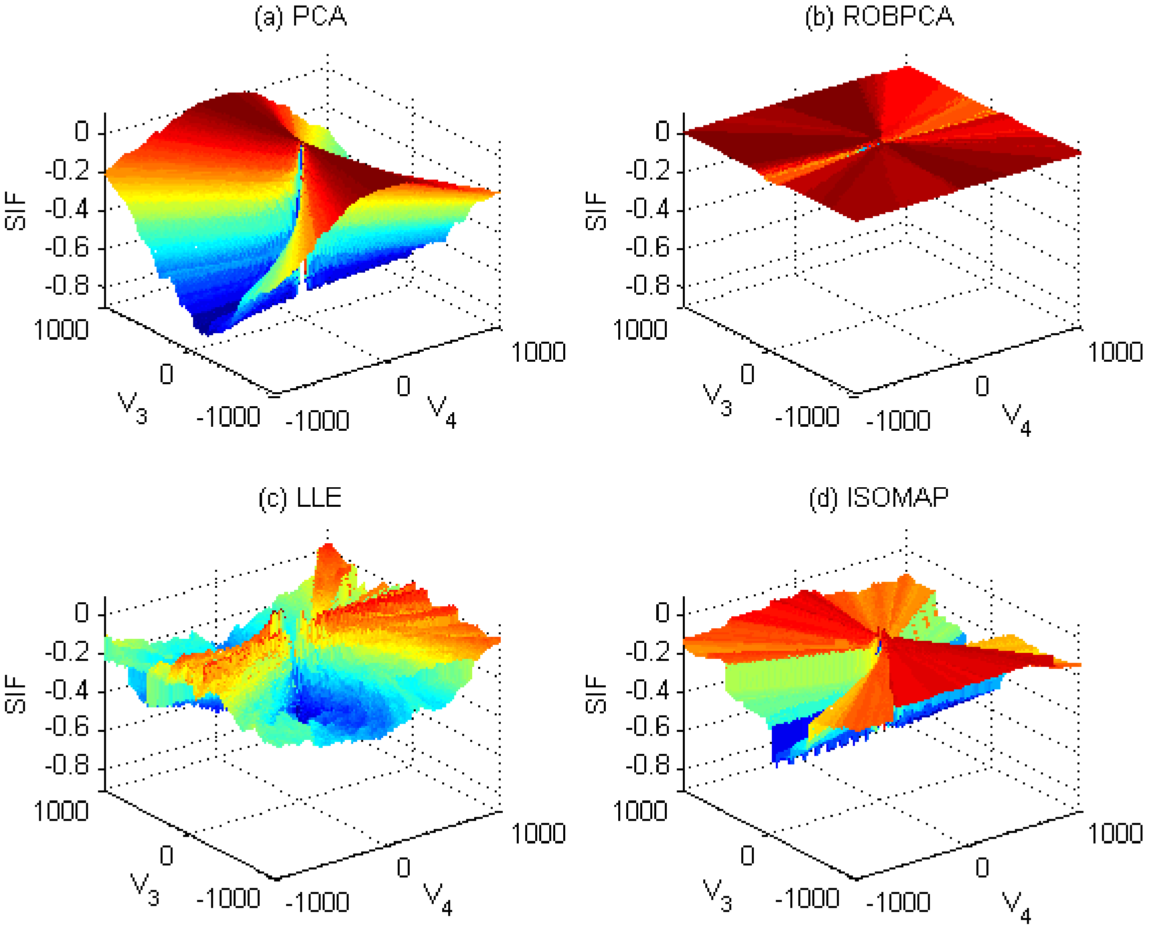

Figure 3.5: Sample influence functions of (a) PCA, (b) ROBPCA, (c) LLE, and (d) ISOMAP.

can not be calculated. The sample influence function in this case provides a supplementary tool in studying the sensitivity of ROBPCA.

In practice, when the observed data $\mathbf{x}_i \in \mathbb{R}^p$, the outlier $\boldsymbol{\omega}$ can be added through p orthogonal directions, and potentially there will be $\binom{p}{2}$ planes to display the sample influence function. Similar to the empirical influence function of Kernel PCA, selecting a most influential plane to plot the sample influence function is also a challenge due to the difficulties of deriving theoretical results. One alternative strategy is to use the symmetry of the sample to find several important directions for plotting. Regardless of any latent curvature structure in the observed data $\mathbf{X}$, we can perform traditional PCA on $\mathbf{X}$. The eigenvectors corresponding to 0 eigenvalues (or close to 0) are symmetric to the sample, and adding a point $\boldsymbol{\omega}$ on the subspace spanned by those eigenvectors should give us similar values of the sample influence function. By doing so, we can reduce the number of planes to be checked. Figure 3.6 shows the sample influence function of ISOMAP on four different planes, based on the sample generated in Example 3.3. In this example, bottom two eigenvectors $\mathbf{v}_4$ and $\mathbf{v}_5$ correspond to eigenvalues that close to 0. As can be seen from the upper left panel (a) and the lower left panel (c), using $\mathbf{v}_4$ or $\mathbf{v}_5$ provides similar graphs. The upper right panel (b) also suggests that adding $\boldsymbol{\omega}$ on the plane spanned by $\mathbf{v}_4$ and $\mathbf{v}_5$ will affect ISOMAP in a similar way.

3.4 Discussion and future work

In this chapter, we developed two influence measures, the empirical influence function and the sample influence function, to analyze the sensitivity of dimension reduction methods. We discussed the strategy of plotting these influence measures, and the possible application of influence measures in outlier detection. There are many research ideas on the sensitivity analysis of dimension reduction that can be studied in the future.

First, we need to further investigate the properties of the empirical influence function for Kernel PCA. It would be practically useful if we can find out a plane on which an added point has the largest influence on a given method. The same problem is also needed to be solved in the sample influence function, where we are interested in searching for the most

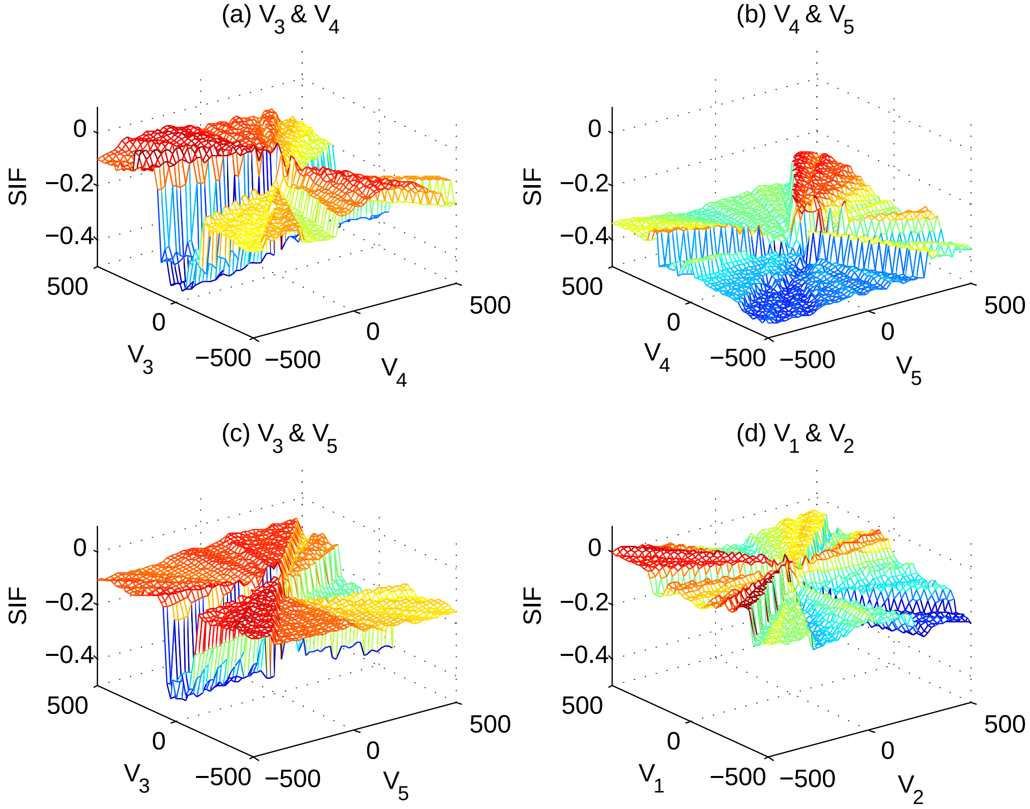

Figure 3.6: Sample influence functions of ISOMAP

influential point (or direction) for a given method.

Another future research topic is to develop other types of robustness measures, such as the breakdown point. Formalization of the concept of "breakdown" is a challenge in dimensionality reduction. One possible solution is to employ the local rank correlation as a criterion. Since the extreme value $G_{_J}(\widehat{\psi}, \mathbf{X}) = 0$ is not always achievable, we can define the breakdown point for a method $\widehat{\psi}$ as

$$\mathrm{BP}(\widehat{\psi}, \mathbf{X}_n) = \min\left\{\frac{m}{n+m} : \inf_{\boldsymbol{\omega}_m \in \otimes_m \mathbb{R}^p} G_{_J}(\widehat{\psi}, \mathbf{X}_{n,m}) \leq c\right\} ,$$

where $\mathbf{X}_{n,m} = [\mathbf{x}_1, \ldots, \mathbf{x}_n, \boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_m]'$ denotes the contaminated sample, and c is a critical value for breakdown. The first challenge in this definition is to find a proper value of c, and the second is to calculate the breakdown point.

Chapter 4

Performance-Weighted Bagging PCA: A New Approach to Robust Principal Component Analysis

4.1 Introduction

4.1.1 Review of robust PCA

As reviewed in Section 1.2.3, principal component analysis (PCA) is a widely-used method in dimensionality reduction. It seeks a linear subspace onto which the projected data have the largest variance. It is typically performed via the eigendecomposition of the sample covariance matrix of the observed data $\mathbf{X}$. Unfortunately, PCA is known to be sensitive to the presence of outlying observations because both the variance and the sample covariance matrix can be heavily influenced by outliers. To overcome the non-robustness, many robust variants of PCA have been proposed over decades. Robust PCA methods can be roughly classified into three groups. In this section, we give a review of these three groups of robust PCA methods (Chenouri et al. [2015]).

The first group of methods achieves the robustness by replacing the sample covariance matrix by a robust alternative. This approach dates back to Maronna [1976] and Campbell [1980] who proposed using affine equivariant M-estimators of the covariance matrix. The M-estimator is shown to be consistent and asymptotically normal under some general assumptions, and has a bounded influence function. However, it has been shown in Donoho [1982] and Rousseeuw and Leroy [1987] that the M-estimators have breakdown value at most $1/p$. Therefore they can only handle a small proportion of outliers when the dimension p of the input data space is sufficiently large. See also Devlin et al. [1981]. Croux and Haesbroeck [2000] proposed using high-breakdown affine equivariant estimators of the covariance matrix such as the minimum volume ellipsoid (MVE) and minimum covariance determinant (MCD) methods of Rousseeuw [1984, 1985] as well as S-estimators of Davies [1987] and Rousseeuw and Leroy [1987]. Although they are very robust, the problem of these methods is that they cannot handle the case when $p > n$, and they are usually computationally expensive. Therefore these methods are applicable only on data with small to moderate dimensions. The fastest algorithms to date can only handle up to about 100 dimensions. See Hubert et al. [2005].

A second approach to make PCA robust is projection pursuit. In this approach, instead of maximizing the variance of projected data, one maximizes a robust measure of dispersion in successive orthogonal directions. Doing this, we bypass the need to robustly estimate the covariance matrix. Some papers on this approach are Li and Chen [1985]; Croux and Ruiz-Gazen [1996]; Hubert et al. [2002a]; Boente et al. [2002]; Maronna [2005]. A fast algorithm is proposed in Croux and Ruiz-Gazen [1996] to overcome the complexity of the traditional projection pursuit method. Hubert et al. [2005] developed a hybrid method which combines advantages of both projection pursuit and high-breakdown covariance estimators. They proposed to first use projection pursuit to reduce the dimensionality to some moderate size and then to apply PCA using MCD estimators of the covariance matrix.

A more recent group of methods consider PCA from the perspective of matrix completion. Suppose the observed data matrix $\mathbf{X}$ is the superposition of a low-rank component and a noisy perturbation, i.e.

$$\mathbf{X} = \mathbf{L}_0 + \mathbf{N}_0\,,$$

where $\mathbf{L}_0$ is a low-rank matrix representing the linear subspace, and $\mathbf{N}_0$ represents the noise. In this setting, the traditional PCA procedure solves a constrained minimization problem

$$\underset{\mathbf{L}}{\operatorname{argmin}}\, \|\mathbf{X} - \mathbf{L}\|_{_F} \quad \text{subject to} \quad \operatorname{rank}(\mathbf{L}) = d\,,$$

where $\|\mathbf{A}\|_{_F} = \sqrt{\operatorname{trace}(\mathbf{A}' \cdot \mathbf{A})}$ is the Frobenius norm of the matrix $\mathbf{A}$. This setup can be also understood as minimizing the reconstruction error. The Frobenius norm is known to be sensitive to outliers, therefore the robustness of PCA could be achieved by replacing the Frobenius norm by a norm less sensitive to outliers. Candès et al. [2011] and Chandrasekaran et al. [2011] independently proposed a robust PCA framework (called principal component pursuit in Candès et al. [2011]). In addition to the assumption that $\mathbf{L}_0$ is low-rank, they assume the noise matrix $\mathbf{N}_0$ is sparse, and the matrix completion is obtained via a convex optimization problem

$$\underset{\mathbf{L},\,\mathbf{N}}{\operatorname{argmin}}\, \{\, \|\, \mathbf{L}\,\|_* + \lambda \,\|\, \mathbf{Z}\,\|_{_1} \,\} \quad \text{subject to} \quad \mathbf{L} + \mathbf{N} = \mathbf{X}\,,$$

where $\|\cdot\|_*$ is the nuclear norm which encourages the low-rankness, and $\|\cdot\|_1$ is the ℓ_1-norm which encourages the sparsity. With a few extra conditions, this problem can be efficiently solved when the dimension is reasonably small (Lin et al. [2009]; Ma et al. [2011]; Candès and Recht [2009]). Many robust PCA variants based on similar idea but with different assumptions have been developed since then, including Zhou et al. [2010]; Zhou and Tao [2011]; Xu et al. [2010]; Wohlberg et al. [2012]; Tang and Nehorai [2011]; Mateos and Giannakis [2010]; Hsu et al. [2011]; Becker et al. [2011]; Podosinnikova et al. [2014]. Although computational challenges still exist when handling large data with high dimensionality, this group of robust PCA variants are now widely applied on many areas such as web data analysis, image and video processing, and background modeling.

In this chapter we will develop a new approach to robustify PCA against outliers from the viewpoint of model averaging, which is potentially compatible with some other robust PCA methods.

4.1.2 Bootstrap aggregating

Model averaging techniques are often employed to improve the accuracy of estimations or predictions in supervised learning problems. One of the most widely known approaches of model averaging is developed by Breiman [1996], called "Bootstrap aggregating" or "Bagging".

Consider a simple regression problem. Suppose we have observed data

$$Z = \{(x_1, y_1), \ldots, (x_n, y_n)\}\,,\; x_i, y_i \in \mathbb{R}\,,$$

and we have a prediction model

$$\widehat{f}(\cdot, Z) : \mathbb{R} \to \mathbb{R}$$

such that based on observed data Z, the outcome y_0 at input x_0 is predicted by

$$\widehat{y}_0 = \widehat{f}(x_0, Z).$$

If we generate a bootstrap sample Z_1 from Z (Efron and Tibshirani [1994]), the prediction based on Z_1 is given by $\widehat{f}(x_0, Z_1)$. The bagging predictor $\widehat{f}_B(x_0)$ is an average over a collection of K predictors based on bootstrap samples $Z_1, \dots, Z_K$ of Z, i.e.

$$\widehat{f}_B(x_0) = \frac{1}{K}\sum_{k=1}^{K} \widehat{f}(x_0, Z_k).$$

In bagging, instead of bootstrap sample, it is also possible to use subsampling, i.e. sampling without replacement. This is known as “Subagging”. Properties of subagging and comparison between bagging and Subagging are discussed in Büchlmann and Yu [2002] and Buja and Stuetzle [2006].

The main purpose of bagging predictor is to reduce the variance of a given procedure. It is stated in Breiman [1996] and Kuncheva [2004] that applying bagging can significantly improve the accuracy of unstable procedures such as regression trees, while stable procedures such as nearest neighbor classifiers are typically not affected much. The performance of bagging, and comparison to other ensemble methods are carried out via experimental studies in Dietterich [2000]. The connection between bagging and Bayes approaches are explored in Friedman et al. [2009].

4.2 Performance-Weighted Bagging PCA

4.2.1 Method description

The basic idea of proposed Performance-Weighted Bagging PCA (PWBPCA) is straightforward. Given a set of p-dimensional data, we employ PCA to obtain a q-dimensional subspace from any subsample of observed data of size larger than q . From the viewpoint of bagging, the subspace produced by traditional PCA based on the entire observed data set can be considered as an average over all subsample subspaces. As shown in Figure 4.1 left panel, the blue line represents the traditional PCA subspace, and black dashed lines represent subsample subspaces.

When outlying points appear in the observed data, although the average subspace can be easily influenced by outliers, only subsamples that contain outliers would produce "bad" subspaces. As shown in Figure 4.1 right panel, red dashed lines represent "bad" subspaces produced by subsamples that contain outliers. If we can find a proper weighting scheme, which assigns lower weights to "bad" subspaces, then we can expect the respective weighted average subspace to be resistant to the presence of outliers.

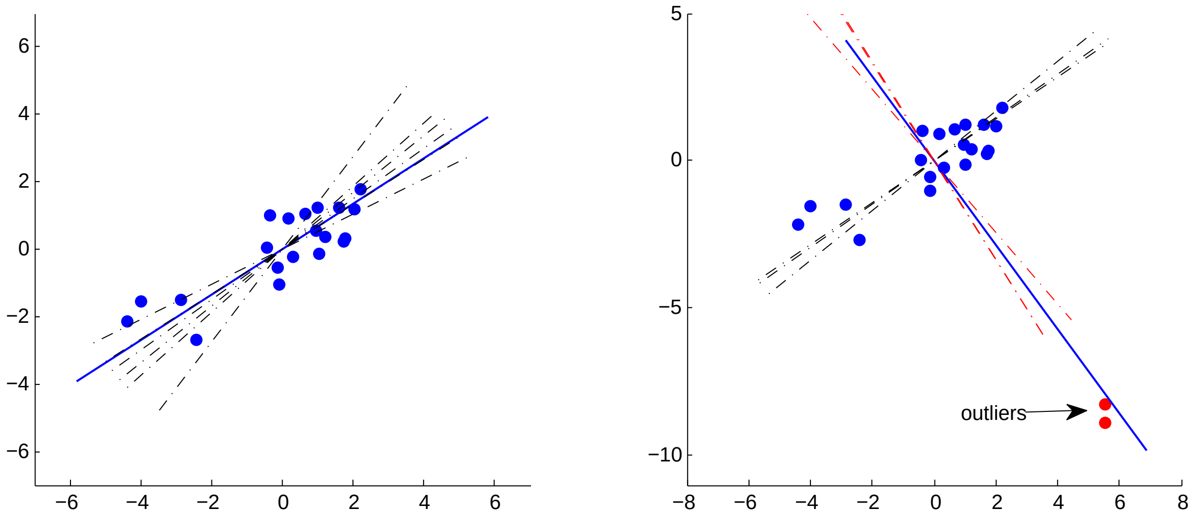

Figure 4.1: PCA from the viewpoint of bagging

First we consider a relatively simpler case, that the intrinsic dimension q, i.e. the number of principal components is given. In this case, the proposed Performance-Weighted Bagging PCA procedure consists of three steps.

(i) Generate K subsamples of size m from original data, where K and m are tuning parameters. For each subsample, obtain the respective q-dimensional subspace.

(ii) Determine the weight $w_{_k}$ for each subsample subspace.

(iii) Obtain the weighted average subspace as the final result.

In step (iii), the weighted average of a set of subspaces can be defined via the distance measure between linear subspaces provided in Definition 3.1.

Definition 4.1. *Weighted average subspace*: Given a set of q-dimensional linear subspaces $\mathcal{S}_{\mathbf{P}_1}, \ldots, \mathcal{S}_{\mathbf{P}_K} \subset \mathbb{R}^p$, and associated weights $w_{_1}, \ldots, w_{_K}$, a distance measure D between subspaces is given by

$$D(\mathcal{S}_{\mathbf{P}_1}, \mathcal{S}_{\mathbf{P}_2}) = [q - \mathrm{trace}(\mathbf{P}_1\, \mathbf{P}_2)]^{1/2} \ ,$$

and the weighted average subspace $\bar{\mathcal{S}}$ with respect to the distance measure D is defined by

$$\bar{\mathcal{S}} = \underset{\mathcal{S}_{\mathbf{P}} \in \mathrm{Gr}(q, \mathbb{R}^p)}{\arg\min} \left\{ \sum_{k=1}^{K} w_{_k} \cdot D^2(\mathcal{S}_{\mathbf{P}_k}, \mathcal{S}_{\mathbf{P}}) \right\} . \tag{4.1}$$

To find the solution to equation (4.1), we can use the following result.

Lemma 4.1. The weighted average subspace $\bar{\mathcal{S}}$ is spanned by the orthonormal columns of the $p \times q$ matrix $\mathbf{U} = \begin{bmatrix} \mathbf{u}_{_1} & \cdots & \mathbf{u}_{_q} \end{bmatrix}$, where $\mathbf{u}_{_i}$ is the eigenvector of the matrix

$$\overline{\mathbf{P}} = \sum_{k=1}^{K} w_{_k} \cdot \mathbf{P}_k$$

corresponding to the i-th largest eigenvalue.

Proof. The solution $\bar{\mathcal{S}}$ to equation (4.1) is uniquely determined by the orthogonal projection matrix $\widehat{\mathbf{P}}$ such that

$$\begin{aligned}\widehat{\mathbf{P}} &= \underset{\text{rank}(\mathbf{P})=q}{\arg\min} \left\{ \sum_{k=1}^{K} w_{_k} \cdot [q - \text{trace}(\mathbf{P}_k\, \mathbf{P})] \right\} \\ &= \underset{\text{rank}(\mathbf{P})=q}{\arg\max} \left\{ \sum_{k=1}^{K} w_{_k} \cdot \text{trace}(\mathbf{P}_k\, \mathbf{P}) \right\} . \end{aligned} \tag{4.2}$$

Solving equation (4.2) is equivalent to solving a $p \times q$ matrix $\widehat{\mathbf{U}}$ such that

$$\widehat{\mathbf{U}} = \underset{\mathbf{U}'\,\mathbf{U}=\mathbf{I}_q}{\arg\max} \left\{ \sum_{k=1}^{K} w_{_k} \cdot \text{trace}(\mathbf{U}' \cdot \mathbf{P}_k \cdot \mathbf{U}) \right\} , \tag{4.3}$$

and then $\widehat{\mathbf{P}}$ can be obtained by $\widehat{\mathbf{P}} = \widehat{\mathbf{U}} \cdot \widehat{\mathbf{U}}'$. Equation (4.3) can be solved by the Lagrange multiplier.

$$\begin{aligned}\mathcal{L}(\mathbf{U}, \lambda) &= \sum_{k=1}^{K} \{w_{_k} \cdot \text{trace}(\mathbf{U}' \cdot \mathbf{P}_k \cdot \mathbf{U})\} + \lambda \cdot (\mathbf{U}'\,\mathbf{U} - \mathbf{I}_q)\,, \\ \frac{\partial \mathcal{L}}{\partial \mathbf{U}} &= 2 \cdot \sum_{k=1}^{K} \{w_{_k} \cdot \mathbf{P}_k \cdot \mathbf{U}\} + 2\lambda \cdot \mathbf{U}. \end{aligned}$$

Let $\overline{\mathbf{P}} = \sum_{k=1}^{K} w_{_k} \cdot \mathbf{P}_k$, and set $\frac{\partial \mathcal{L}}{\partial \mathbf{U}} = 0$, we have

$$\overline{\mathbf{P}} \cdot \mathbf{U} = \lambda \cdot \mathbf{U}\,. \tag{4.4}$$

Clearly, equation (4.4) implies that the orthonormal columns $\left\{\mathbf{u}_{_1}, \ldots, \mathbf{u}_{_q}\right\}$ of the matrix $\widehat{\mathbf{U}}$ are eigenvectors of $\overline{\mathbf{P}}$ corresponding to the q largest eigenvalues. □

A key step in PWBPCA procedure is step (ii), determining the weight for each subspace. We want to lower the effect of "bad" subspaces via a proper weighting scheme to achieve robustness. Intuitively, when we project the data onto a subspace, if the geometric structure of the data is not well preserved, then we would call the subspace "bad". Therefore a natural choice of the weight function is local rank correlations proposed in Chapter 2. Given a subspace $\mathcal{S}_k$ spanned by orthogonal columns of $\mathbf{U}_k$, denote

$$\widehat{\psi}_{k} : \mathbf{X} \longmapsto \mathbf{X} \cdot \mathbf{U}_k \,.$$

The low-dimensional representation projected onto $\mathcal{S}_k$ is denoted by $\widehat{\mathbf{Y}}_k = \widehat{\psi}_{k}(\mathbf{X})$. Using the notation in Section 2.3, the weight w_{k} for the subspace $\mathcal{S}_k$ is obtained by

$$w_{k} = G_{J}(\widehat{\psi}_{k}, \mathbf{X}) = \frac{1}{n}\sum_{i=1}^{n} \Gamma_{J}(i, \mathbf{X}, \widehat{\psi}_{k}(\mathbf{X}))\,,$$

where one can choose Γ_{J} to be either one of ρ_{J}^{O}, ρ_{J}^{I}, τ_{J}^{O} or τ_{J}^{I}. It is also possible to choose the weight w_{k} to be a combination of local rank correlations, such as $(\rho_{J}^{O} + \rho_{J}^{I})/2$ or $\sqrt{\rho_{J}^{O} \cdot \rho_{J}^{I}}$. In numerical examples in the following sections, unless specified otherwise we choose the weight w_{k} to be

$$w_{k} = \frac{1}{n}\sum_{i=1}^{n} \rho_{6}^{O}(i, \mathbf{X}, \widehat{\psi}_{k}(\mathbf{X}))\,.$$

Note that the weight w_{k} does not need to be normalized, i.e. it is not necessary to have $\sum_k w_{k} = 1$. Because the normalization

$$\widetilde{\mathbf{P}} = \sum_{k=1}^{K} \left(\frac{w_{k}}{\sum_k w_{k}} \right) \cdot \mathbf{P}_k = \overline{\mathbf{P}}/(\sum_k w_{k})$$

is simply a scaling transformation on the matrix $\overline{\mathbf{P}}$, and scaling does not change the eigenvector of $\overline{\mathbf{P}}$. Later in Section 4.2.3, we will further discuss more possible ways to choose the weighting scheme in Performance-Weighted Bagging PCA.

Algorithm 1 provides the procedure of Performance-Weighted Bagging PCA, with a given number of components q.

Algorithm 1 Performance-Weighted Bagging PCA (with q known)

```
Input: $\mathbf{X} = [\mathbf{x}_1 \cdots \mathbf{x}_n]'$, $q$, $m$, $K$
Generate $K$ subsamples $\mathbf{X}_1, \ldots, \mathbf{X}_K$ of size $m$ from the original data $\mathbf{X}$ by resampling without replacement.
for $k = 1$ to $K$ do
    Perform traditional PCA procedure with $\mathbf{X}_k$, and obtain a $p \times q$ column orthogonal matrix $\mathbf{U}_k$ and corresponding projection matrix $\mathbf{P}_k = \mathbf{U}_k \, \mathbf{U}_k'$.
    Obtain $\widehat{\mathbf{Y}}_k = \mathbf{X}{\cdot}\mathbf{U}_k$ and calculate the weight $w_k$ for $\mathbf{P}_k$ based on local rank correlation between $\mathbf{X}$ and $\widehat{\mathbf{Y}}_k$.
end for
Calculate the weighted average
    $$\overline{\mathbf{P}} = \sum_{k=1}^{K} w_k \cdot \mathbf{P}_k \, .$$
Select top $q$ eigenvectors of $\overline{\mathbf{P}}$ to form $\widehat{\mathbf{U}} = \left[\widehat{\mathbf{u}}_1 \cdots \widehat{\mathbf{u}}_q\right]$.
Obtain $q$-dimensional representation by $\widehat{\mathbf{Y}} = \mathbf{X} \cdot \widehat{\mathbf{U}}$.
Output: $\widehat{\mathbf{Y}} = [\widehat{\mathbf{y}}_1 \cdots \widehat{\mathbf{y}}_n]'$
```

4.2.2 Remarks

The first remark is on the computational complexity of the proposed algorithm. The algorithm consists of three major steps, and the complexity of each step scales as

- Perform PCA on each subsample: $O(mp^2 + p^3)$.
- Calculate the weight: $O(n^2 p)$.
- Obtain weighted average subspace: $O(Kp^2)$.

When m is fixed, the total complexity scales as $O(K(n^2 p + p^3))$. When dealing with a large value of p, solving PCA in the first step can be done in its dual form, i.e.

- Perform eigendecomposition on $\mathbf{X}_k \cdot \mathbf{X}_k'$ and obtain a $q \times q$ eigenvalue matrix $\mathbf{\Lambda}_k$ and a $m \times q$ eigenvector matrix $\mathbf{V}_k$. This step scales as $O(m^2 p + m^3)$.
- Obtain $\mathbf{U}_k = \mathbf{X}_k' \cdot \mathbf{V}_k \cdot \mathbf{\Lambda}_k^{-1/2}$. This step scales as $O(q^3 m^2 p)$.
- Obtain $\mathbf{P}_k = \mathbf{U}_k \, \mathbf{U}_k'$. This step scales as $O(q^2 p^2)$.

Note that when the number of components q is small, the total complexity can be reduced to $O(K(n^2 p + p^2))$. The computational complexity of PWBPCA does not increase dramatically as the dimension p increases, which is an advantage compare to some traditional methods. For example, the complexity of robust PCA based on MCD-estimator scales as $O(n^{p(p+3)/2})$ (Bernholt and Fischer [2004]).

The second remark is on tuning parameters m and K. A typical choice of the subsample size m in traditional bagging procedures is a fraction of n, i.e. $m = a \cdot n$ with $0 < a < 1$. It is suggested in Büchlmann and Yu [2002] that a reasonable choice of a is $a = 1/2$.

In our Performance-Weighted Bagging PCA procedure, the subsample size m controls the trade-off between efficiency and robustness. Intuitively, when there is no outlier in the input data, a larger value of m will make better use of the sample and therefore produce a better result in the sense of efficiency. On the other hand, when the input data contain outliers, a relatively large value of m does not seem to be a good choice. The key idea in the Performance-Weighted Bagging PCA is that via subsampling, a decent part of subsamples will produce good subspaces, while subsamples containing outliers can be handled by down-weighting. However, as m increases, the chance that a subsample contains at least one outlier also increases. For example, if we have a sample with size $n = 50$, and 10% of the data are outliers, choosing $m = 20$ would lead to the result that roughly 93.27% of subsamples contain at least one outlier. In this case, although being down-weighted, cumulatively "bad" subspaces can still heavily affect the final result. This argument is illustrated in Example 4.1 below.

Example 4.1. *Effect of subsample size m:* We generate a sample of size $n = 100$ from a contaminated multivariate normal distribution,

$$\begin{aligned}
&\mathbf{x}_i \sim (1-\epsilon)\cdot\mathcal{N}_5(\mathbf{0},\mathbf{\Sigma}) + \epsilon\cdot\mathcal{N}_5(\boldsymbol{\mu}_{1}, 0.1\cdot\mathbf{I}_5),\ i = 1,\ldots,100,\\
&\boldsymbol{\mu}_{1} = 50\cdot\mathbf{e}_{5},\\
&\mathbf{\Sigma} = \mathrm{diag}([8,6,4,0.5,0.1])\,.
\end{aligned}$$

where $\mathbf{I}$ is the identity matrix and $\mathbf{e}_{i}$ is a $p\times 1$ vector whose i-th element equals 1 and all other elements equal 0. In this case, the intrinsic dimension $q = 3$, and the true subspace $\mathcal{S}$ is spanned by $\{\mathbf{e}_{1},\mathbf{e}_{2},\mathbf{e}_{3}\}$. We apply Performance-Weighted Bagging PCA with different values of subsample size m, different values of contamination level ϵ, and number of subsamples $K = 50$. At each pair of $(m,\,\epsilon)$, the procedure is repeated 1000 times, and the mean square distance (MSD) between the true subspace and the subspace obtained from Performance-Weighted Bagging PCA is evaluated by,

$$\mathrm{MSD} = \frac{1}{1000} \sum_{i=1}^{1000} D^2(\mathcal{S}, \widehat{\mathcal{S}}^{(i)}_{_{BPCA}})\,.$$

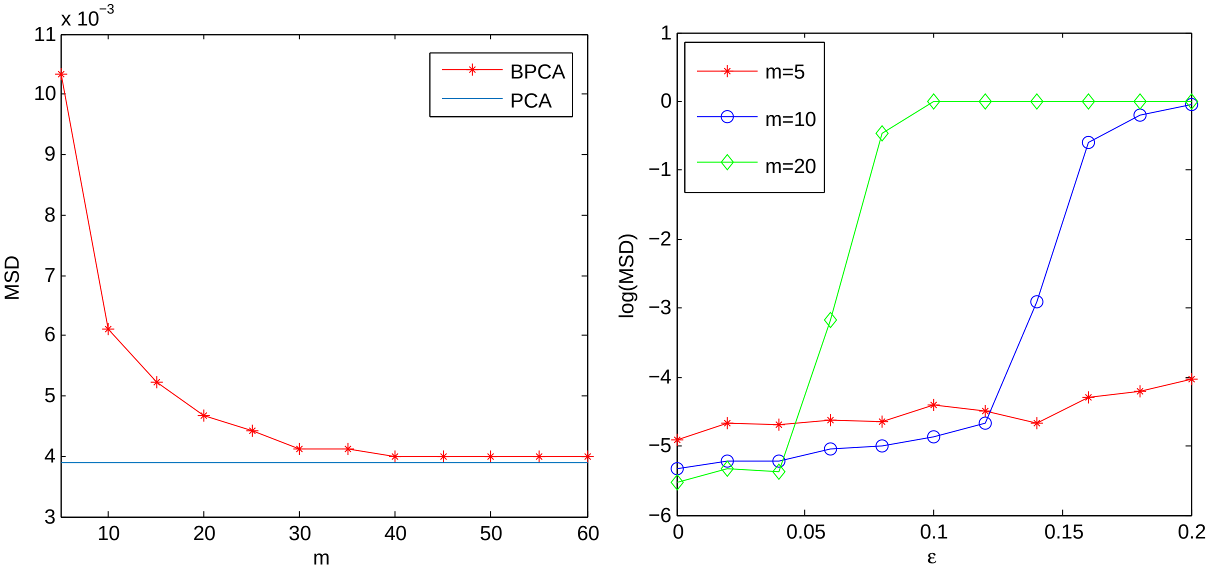

Figure 4.2: Effect of subsample size m $(K = 50)$

Figure 4.2 left panel plots the mean square distance as a function of m when $\epsilon = 0$ (no contamination). The blue solid line represents the expected value of the squared distance between true subspace and traditional PCA subspace (Crone and Crosby [1995]),

$$\mathrm{E}(D^2(\mathcal{S}, \widehat{\mathcal{S}}_{_{PCA}})) = \sum_{i=1}^{q} \sum_{j=q+1}^{p} \frac{\lambda_{_i} \cdot \lambda_{_j}}{n(\lambda_{_i} - \lambda_{_j})^2}\,,$$

where $\lambda_{_i}$ is the i-th largest eigenvalue of $\mathbf{\Sigma}$. As can be seen, as m increases, the performance of Performance-Weighted Bagging PCA, in terms of the MSD, is getting better. Figure 4.2 right panel plots the mean square distance (in log scale) as a function of ϵ. Clearly, a smaller value of m is more resistant to the presence of outliers.

The number of subsamples K controls the trade-off between the efficiency and the computational cost. When m is large, for instance $m = n/2$, it seems like the value of K does not have much effect. However, when m is small, a larger value of K could improve the efficiency of Performance-Weighted Bagging PCA (as illustrated in Figure 4.3). Using the same setting in Example 4.1, Figure 4.3 plots the mean square distance as a function of K with $m = 5$. As can be seen, the mean square distance decreases as K increases. On the other hand, as we mentioned above, the computational complexity for Performance-Weighted Bagging PCA is linear in K. Therefore a large value of K would increase the computational cost.

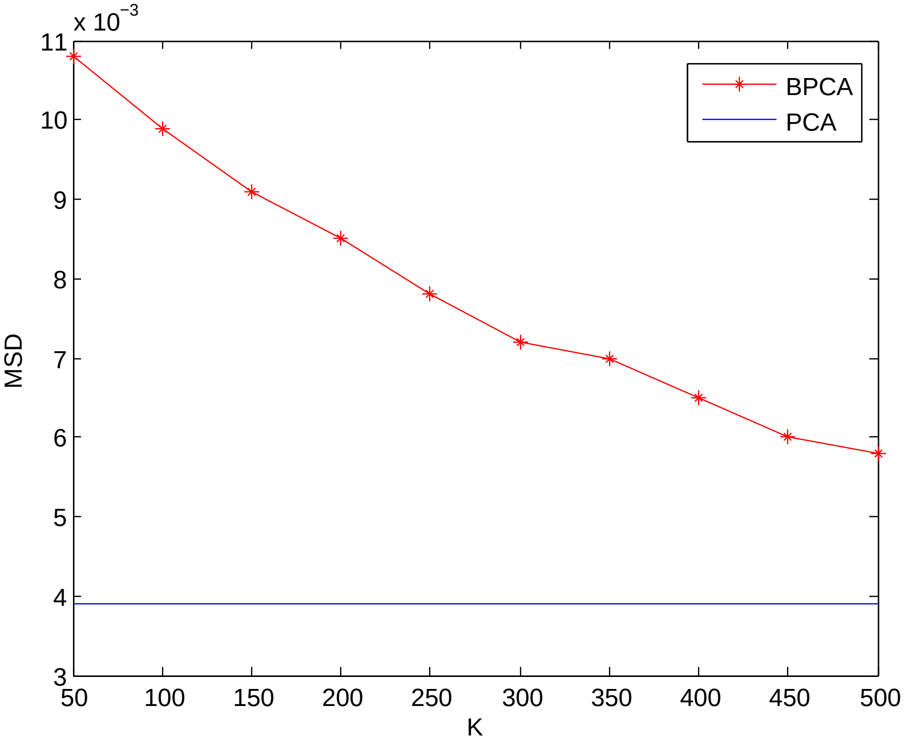

Figure 4.3: Effect of number of subsample K ($m = 5$)

In practice, a reasonable strategy is to choose a small value of the subsample size m, for example $m = q + 1$, to achieve robustness, and relatively large values of K to remedy the loss of efficiency. The increase in the computational cost due to a large K can be made up by parallel computing.

Note that the idea of applying bagging in PCA has been used a few times in the literature. Gabrys et al. [2006] developed a method called PCA ensembles. The proposed procedure starts from generating subsamples from the original dataset and performs PCA on each subsample to obtain respective top eigenvectors. Then these eigenvectors are clustered, and the average of the largest cluster is calculated as a principal component. Leng et al. [2014] proposed bagging in combination with PCA to learn effective binary codes. A short code is generated by performing PCA on each subsample of training set, and then a set of short codes is concatenated into one piece of long code. The key difference between these previous bagging methods and the proposed Performance-Weighted Bagging PCA lies in the use of weighting scheme. A proper choice of weighting scheme adjusts the contribution of each subsample subspace according to its performance, making the weighted average subspace more robust than unweighted average subspace.

It is also worth mentioning that another possible way to perform PCA from the perspective of model averaging is proposed in Liski et al. [2012]. The idea is to perform several different PCA variants, and take the average of resulting subspaces to obtain a compromise estimate. For each PCA variant, the entire input data $\mathbf{X}$ are used to calculate the corresponding subspace. The main difference between this method and our Performance-Weighted Bagging PCA is that the Performance-Weighted Bagging PCA is averaging over a set of subsamples, but for each subsample only traditional PCA is performed. The robustness of the Performance-Weighted Bagging PCA is gained from a proper weighting scheme.

4.2.3 Weighting scheme

The robustness of Performance-Weighted Bagging PCA is achieved by down-weighting subsamples containing outliers. A subsample subspace is an estimate of the underlying subspace, and a candidate weight function w_k should reflect the accuracy of this estimation. Besides the performance measure, it is also possible to obtain a proper weighting scheme from the loss function in other robust PCA methods. In this section, we will discuss other possible ways to select the weight w_k, which will potentially make connections between Performance-Weighted Bagging PCA and other robust PCA methods.

If we formulate PCA from the viewpoint of maximizing the variance of projected data, one way to robustify PCA, as reviewed in Section 4.1.1, is via projection pursuit. The projection pursuit approach searches a subspace that maximizes a robust measure of dispersion of projected data. In other words, a robust measure of dispersion is used to evaluate the accuracy of an estimated subspace. Thus, in the performance-weighted bagging framework, the weight w_k of a subsample subspace $\mathcal{S}_k$ can be chosen as

$$w_k = \widetilde{\sigma}(\widehat{\psi}_k(\mathbf{X}))\,,$$

where $\widetilde{\sigma}(\cdot)$ is a robust measure of dispersion. The subspace that preserves less variability, in terms of $\widetilde{\sigma}$, will receive a lower value of weight. For example, a possible choice of $\widetilde{\sigma}$ is the sum of robust singular values of $\widehat{\psi}_k(\mathbf{X})$ (Ammann [1993]).

If we formulate PCA from the viewpoint of minimizing the reconstruction error, PCA can be robustified by minimizing a robust measure of reconstruction error. Each subsample subspace reconstructs the input data by $\widehat{\mathbf{X}}^{(k)} = \mathbf{X} \cdot \mathbf{P}_k$, and the weight w_k in this case can be chosen as

$$w_k = \frac{1}{\mathrm{Rerr}(\mathbf{X}, \widehat{\mathbf{X}}^{(k)})}\,,$$

where $\mathrm{Rerr}(\mathbf{X}, \widehat{\mathbf{X}}^{(k)})$ is a robust measure of reconstruction error between $\mathbf{X}$ and $\widehat{\mathbf{X}}^{(k)}$. For example, a possible choice of robust reconstruction error is provided in Podosinnikova et al. [2014].

In these two viewpoints, the advantage of performing weighted bagging instead of solving the optimization problem directly is in computation. Some robust measures, for instance median absolute deviation, is difficult to optimize, but is easy to calculate. Therefore, the optimal subspace characterized by these robust measures can be computationally challenging. Performance-Weighted Bagging PCA in these cases provides an alternative way by incorporating the chosen robust measure in the weight $w_{_k}$. These two examples suggest the potential of connecting other robust PCA methods with the Performance-Weighted Bagging framework.

Another possible extension on the weighting scheme is that one can employ a monotonic increasing transformation on the weight $w_{_k}$,

$$\widetilde{w}_{_k} = g(w_{_k}) \, ,$$

where $g(\cdot)$ is a monotonic increasing function. The purpose of employing $g(\cdot)$ is to magnify the different contributions between “good” subspaces and “bad” subspaces.

This strategy can be useful when the contamination level ϵ is large. In this case, most of subsamples will contain outliers and produce “bad” subspaces. As we discussed in Section 4.2.2, one way to deal with this situation is to choose a small value of subsample size m to make Performance-Weighted Bagging PCA more resistant to outliers. On the other hand, however, m has to be larger than q to fit a q-dimensional subspace. Therefore, if one wants to choose a relatively large value of m without losing the robustness, it could be done via a proper transformation $g(\cdot)$. Figure 4.4 shows some choices of $g(\cdot)$.

Consider the contaminated multivariate normal distribution in Example 4.1. We apply

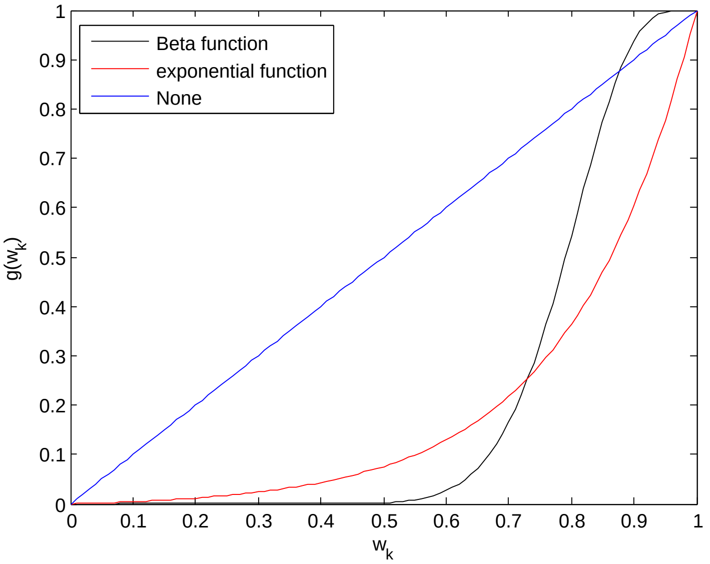

Figure 4.4: Monotonic transformation of weights w_k.

Performance-Weighted Bagging PCA with $m = 10$ and $K = 100$. The weight w_k is first calculated by the local rank correlation. We then employ an exponential transformation

$$g(w_k) = \mathrm{e}^{5 \cdot w_k} \ ,$$

to obtain a new weighting scheme. Figure 4.5 plots the histogram of weights w_k in the left panel and $g(w_k)$ (after normalization) in the right panel, when the contamination level $\epsilon = 0.2$. Clearly, after the transformation, subsamples with outliers are further down-weighted, and nearly have no contributions to the final average. Figure 4.6 shows the mean square distance (in log scale) between the true subspace and the weighted average subspaces as functions of ϵ. As can be seen, this exponential transformation on the weights makes Performance-Weighted Bagging PCA more resistant to outliers in this example.

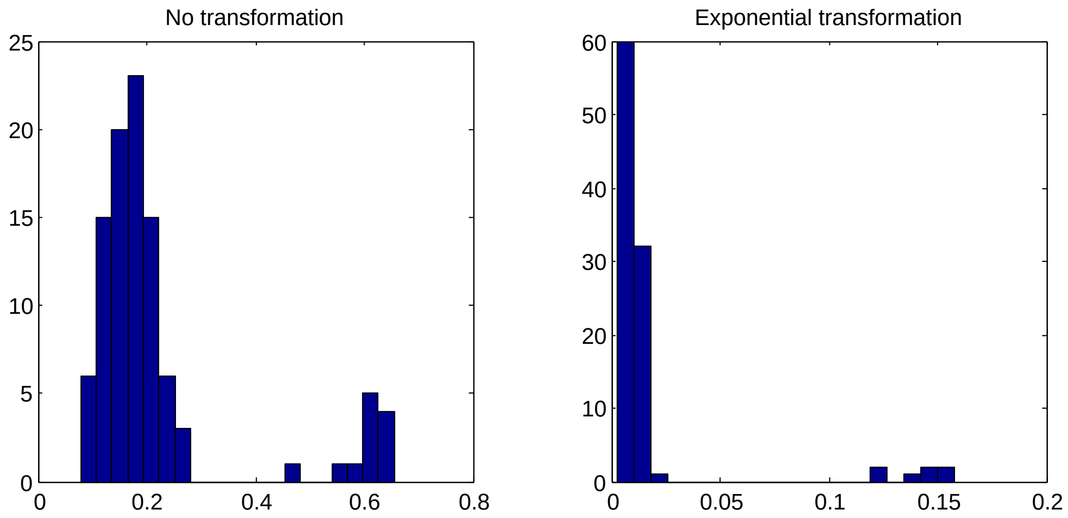

Figure 4.5: Histogram of weights w_k with and without transformation ($\epsilon = 0.2$).

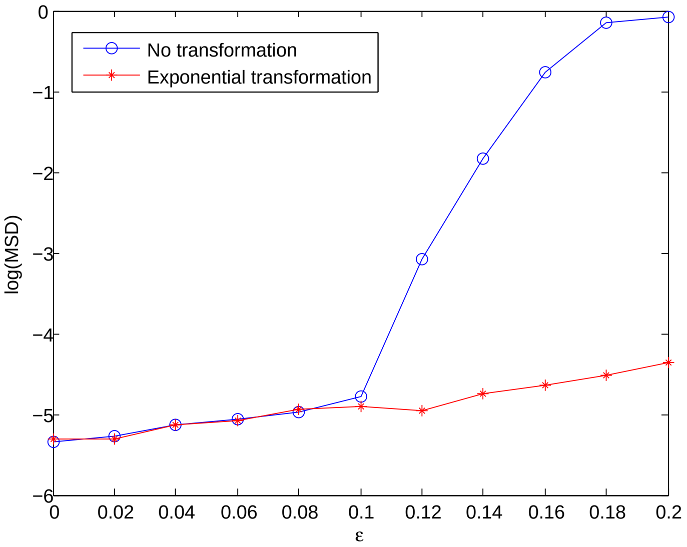

Figure 4.6: Effect of transformation on the weight function.

4.3 Selection of the number of components

Determining the number of principal components q is an important problem in PCA. A variety of stopping rules to estimate q has been proposed over decades. A stopping rule $q(\cdot) : \mathbb{R}^p \to \mathbb{N}$ is a mapping which takes the observed high-dimensional data $\mathbf{X}$ as the input, and produce an estimate $\widehat{q}$ as the output. Some reviews and comparisons of different stopping rules can be found in Ferré [1995]; Jackson [2005]; Peres-Neto et al. [2005].

In Performance-Weighted Bagging PCA, when the dimension q is not given but to be estimated, one obvious choice is to obtain an estimate $\widehat{q}$ based on the entire sample and plug $\widehat{q}$ into algorithm 1. Another choice is to estimate q during the procedure of bagging. To apply this idea, the first task is to determine the dimension for each subsample subspace. Our strategy is to adopt any feasible stopping rule and individually estimate the dimension q_k of each subsample subspace $\mathcal{S}_k$. We allow the dimension q_k to be different from each other. After obtaining a set of subsample subspaces with different dimensions, our second task is to define the weighted average subspace. The concept of distance has been extended by Liski et al. [2012] to measure the dissimilarity between two subspaces with arbitrary dimensions.

Definition 4.2. *Distance between subspaces with arbitrary dimensions*: Given two linear subspaces $\mathcal{S}_{\mathbf{P}_1}$, $\mathcal{S}_{\mathbf{P}_2}$ in $\mathbb{R}^p$, with $\text{rank}(\mathbf{P}_1) = q_1$ and $\text{rank}(\mathbf{P}_2) = q_2$. The weighted distance between $\mathcal{S}_{\mathbf{P}_1}$ and $\mathcal{S}_{\mathbf{P}_2}$ is define by

$$D_h\left(\mathcal{S}_{\mathbf{P}_1}, \mathcal{S}_{\mathbf{P}_2}\right) = \frac{1}{\sqrt{2}} \left\| h(q_1)\mathbf{P}_1 - h(q_2)\mathbf{P}_2 \right\|_F \tag{4.5}$$

$$= \left[\frac{h^2(q_1)q_1 + h^2(q_2)q_2}{2} - h(q_1)h(q_2) \cdot \text{trace}(\mathbf{P}_1\,\mathbf{P}_2) \right]^{1/2}, \tag{4.6}$$

where $h(\cdot)$ is a positive weight function.

Some possible choices of $h(\cdot)$ include

(a) $h(q) = 1$,

(b) $h(q) = 1/q$,

(a) $h(q) = 1/\sqrt{q}$.

The weight function in (a) is the direct generalization of distance measure by Crone and Crosby [1995], while (b) and (c) standardize the projection matrix. The interpretation and properties of the weighted distance measure are discussed in Liski et al. [2012].

Similar to Lemma 4.1, the weighted average subspace can be defined by the weighted distance measure provided in Definition 4.2.

Definition 4.3. Given a set of subspaces $\mathcal{S}_{\mathbf{P}_1}, \ldots, \mathcal{S}_{\mathbf{P}_K} \subset \mathbb{R}^p$ with ranks $q_1, \ldots, q_K$, and associated weights $w_1, \ldots, w_K$, the weighted average subspace $\bar{\mathcal{S}}$ with respect to the weighted distance measure D_h is defined by

$$\bar{\mathcal{S}} = \underset{\mathcal{S}_{\mathbf{P}} \subset \mathbb{R}^p}{\arg\min} \left\{ \sum_{k=1}^{K} w_k \cdot D_h^2(\mathcal{S}_{\mathbf{P}_k}, \mathcal{S}_{\mathbf{P}}) \right\} . \tag{4.7}$$

To find the weighted average subspace $\bar{\mathcal{S}}$, we can use the following lemma.

Lemma 4.2. Given an integer $1 \leq q \leq p$, the q-dimensional weighted average subspace $\bar{\mathcal{S}}_{(q)}$ is defined as

$$\bar{\mathcal{S}}_{(q)} = \underset{\mathcal{S}_{\mathbf{P}} \in \mathrm{Gr}(q, \mathbb{R}^p)}{\arg\min} \left\{ \sum_{k=1}^{K} w_k \cdot D_h^2(\mathcal{S}_{\mathbf{P}_k}, \mathcal{S}_{\mathbf{P}}) \right\} .$$

The solution $\bar{\mathcal{S}}_{(q)}$ is spanned by the orthonormal columns of the $p \times q$ matrix $\mathbf{U} = \begin{bmatrix} \mathbf{u}_1 & \cdots & \mathbf{u}_q \end{bmatrix}$, where $\mathbf{u}_i$ is the eigenvector of the matrix

$$\overline{\mathbf{P}} = \sum_{k=1}^{K} w_k h(q_k) \mathbf{P}_k$$

corresponding to the i-th largest eigenvalue. The solution $\bar{\mathcal{S}}$ to equation (4.7) is then obtained by

$$\bar{\mathcal{S}} = \operatorname*{arg\,min}_{1\le q\le p} \left\{ \sum_{k=1}^{K} w_{_k} \cdot D_h^2(\mathcal{S}_{\mathbf{P}_k}, \bar{\mathcal{S}}_{(q)}) \right\} .$$

Since the subsample size is chosen to be m, the dimension of subsample subspace $q_{_k}$ takes value between 1 and $m-1$. Therefore, to obtain $\bar{\mathcal{S}}$, we only need to search q from 1 to $m-1$. In other words, if we define the *weighted mean square distance* between $\bar{\mathcal{S}}_{(q)}$ and each subsample subspace as

$$\text{WMSD}(q) = \frac{1}{K}\sum_{k=1}^{K} w_{_k} \cdot D_h^2(\mathcal{S}_{\mathbf{P}_k}, \bar{\mathcal{S}}_{(q)}) \, ,$$

the final estimated number of components $\widehat{q}$ is the one that minimizes the weighted mean square distance, i.e.

$$\widehat{q} = \operatorname*{arg\,min}_{1\le q<m} \{\text{WMSD}(q)\} \, .$$

The success of the proposed procedure obviously depends on the stopping rule we choose. However, one advantage of selecting q during bagging is that it not only robustifies PCA, but also robustifies the stopping rule against the presence of outliers. Because subsamples with outliers will be down-weighted, as long as the dimension of most subsamples without outliers can be estimated correctly from the chosen stopping rule, we can expect the final $\widehat{q}$ to be correct. Therefore even applying a non-robust stopping rule to subsamples will result in a robust final estimate $\widehat{q}$. Also note that the function $h(q)$ is used to make subspaces with different dimensions comparable. The choice of $h(q)$ does not affect the final estimate $\widehat{q}$. In all the simulations and real data analysis, we choose $h(q) = 1$ unless specified otherwise.

Algorithm 2 provides the procedure of PWBPCA, with the number of components q unknown.

Algorithm 2 Performance-Weighted Bagging PCA (with q unknown)

```
Input: $\mathbf{X} = [\mathbf{x}_1 \cdots \mathbf{x}_n]'$, $q(\cdot)$, $m$, $K$
Generate $K$ subsamples $\mathbf{X}_1, \ldots, \mathbf{X}_K \subset \mathbf{X}$ of size $m$ from the original data by resampling without replacement.
for $k = 1$ to $K$ do
    Perform traditional PCA procedure with $\mathbf{X}_k$, and obtain a $p \times q_k$ column orthogonal matrix $\mathbf{U}_k$ and corresponding projection matrix $\mathbf{P}_k = \mathbf{U}_k\,\mathbf{U}_k'$, where $q_k$ is determined by the stopping rule $q_k = q(\mathbf{X}_k)$.
    Obtain $\widehat{\mathbf{Y}}_k = \mathbf{X}\cdot\mathbf{U}_k$ and calculate the weight $w_k$ for $\mathbf{P}_k$ based on local rank correlation between $\mathbf{X}$ and $\widehat{\mathbf{Y}}_k$.
end for
Calculate the weighted average
    $$\overline{\mathbf{P}} = \sum_{k=1}^{K} w_k \cdot \mathbf{P}_k \,.$$
Select top $\widehat{q}$ eigenvectors of $\overline{\mathbf{P}}$ to form $\widehat{\mathbf{U}}$, such that
    $$\widehat{\mathbf{U}} = \underset{1\le q\le p}{\arg\min} \left\{ \sum_{k=1}^{K} w_k \cdot \left\| \widehat{\mathbf{U}}_{(q)}\,\widehat{\mathbf{U}}_{(q)}' - \mathbf{P}_k \right\|_F^2 \right\} ,$$
    where $\widehat{\mathbf{U}}_{(q)} = \left[\widehat{u}_1 \;\cdots\; \widehat{u}_q\right]$ .
Obtain $\widehat{q}$-dimensional representation by $\widehat{\mathbf{Y}} = \mathbf{X} \cdot \widehat{\mathbf{U}}$.
Output: $\widehat{\mathbf{Y}} = [\widehat{\mathbf{y}}_1 \cdots \widehat{\mathbf{y}}_n]'$.
```

In the following example, we illustrate how this procedure selects the number of components.

Example 4.2. We generate the input data $\mathbf{X}$ in $\mathbb{R}^{10}$ from a contaminated multivariate normal distribution,

$$\begin{aligned}
&\mathbf{x}_i \sim (1-\epsilon)\cdot\mathcal{N}_{10}(\mathbf{0},\mathbf{\Sigma}) + \epsilon\cdot\mathcal{N}_{10}(\boldsymbol{\mu}_1, 0.1\cdot\mathbf{I}_{10}),\ i = 1,\dots,100\\
&\boldsymbol{\mu}_1 = 100\cdot\mathbf{e}_{10}\\
&\mathbf{\Sigma} = \mathrm{diag}([8,6,4,\underbrace{0.1,\dots,0.1}_{7}])\,.
\end{aligned}$$

In this case, the true number of components is $q = 3$. We apply Performance-Weighted Bagging PCA in algorithm 2 with $m = 10$, $K = 500$, and we choose the stopping rule to be

$$q_k = q(\mathbf{X}_k) = \min\left\{q : \frac{\sum_{j=1}^{q}\lambda_{(k),j}}{\sum_{j=1}^{p}\lambda_{(k),j}} > 0.9\right\}\,,$$

where $\lambda_{(k),j}$ is the j-th largest eigenvalue of the covariance matrix of $\mathbf{X}_k$. In other words, for each subsample subspace, the dimension q_k is chosen to be the smallest number of components that accounts for more than 90% of the total variance of the subsample. We examine two cases where $\epsilon = 0$ and $\epsilon = 0.2$. Figure 4.7 plots the histogram of the estimated subspace dimension q_k and Figure 4.8 plots the weighted mean square distance as a function of q. Note that when there are 20% of outliers in the input data, most of subsamples contain outliers (roughly 90%). Although the stopping rule we chose is non-robust and it provides a wrong estimate for these subsample subspaces, i.e. $q_k = 1$ (as shown in the right panel in Figure 4.7), the proposed procedure still selects the correct number of components, i.e. $\widehat{q} = 3$ (as shown in the right panel in Figure 4.8).

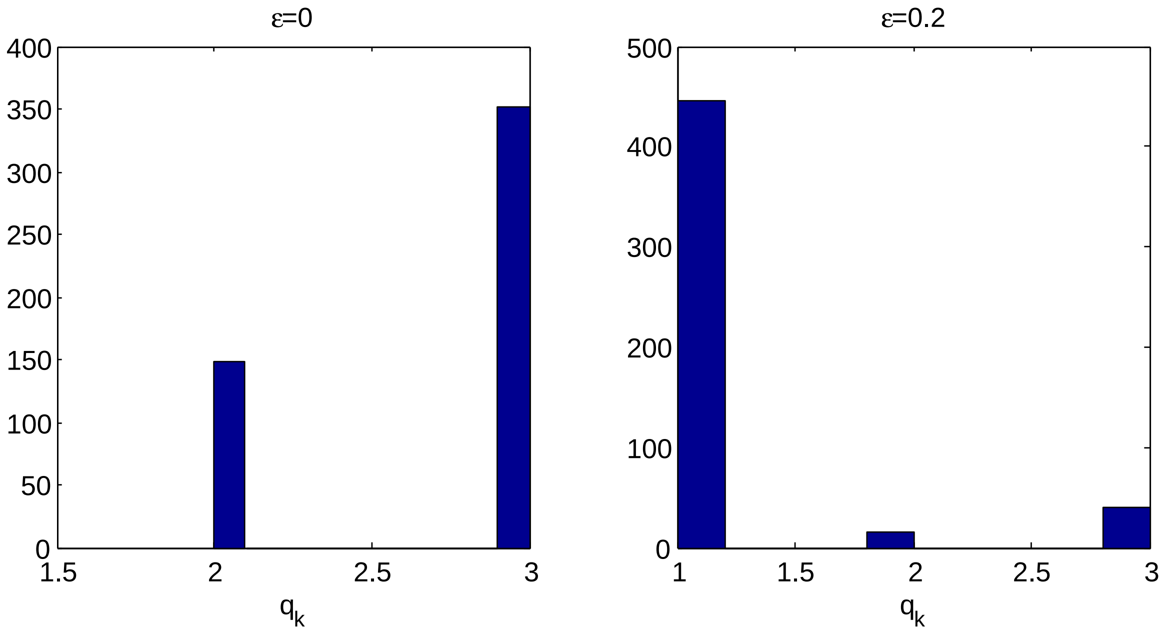

Figure 4.7: Histogram of estimated subspace dimension q_k

4.4 Simulation study

In this section, we conduct a simulation study to compare the performance and the robustness of Performance-Weighted Bagging PCA. We generate the input data $\mathbf{X}$ from a contaminated multivariate normal distribution,

$$\mathbf{x}_i \sim (1-\epsilon)\cdot\mathcal{N}_p(\mathbf{0},\mathbf{\Sigma}) + \epsilon\cdot\mathcal{N}_p(\boldsymbol{\mu}_1, 0.1\cdot\mathbf{I}_p),\ i = 1,\ldots,100.$$

with different values of p, $\mathbf{\Sigma}$, ϵ, and $\boldsymbol{\mu}_1$. We apply four different robust PCA methods

- Performance-Weighted Bagging PCA (PWBPCA),
- ROBPCA,

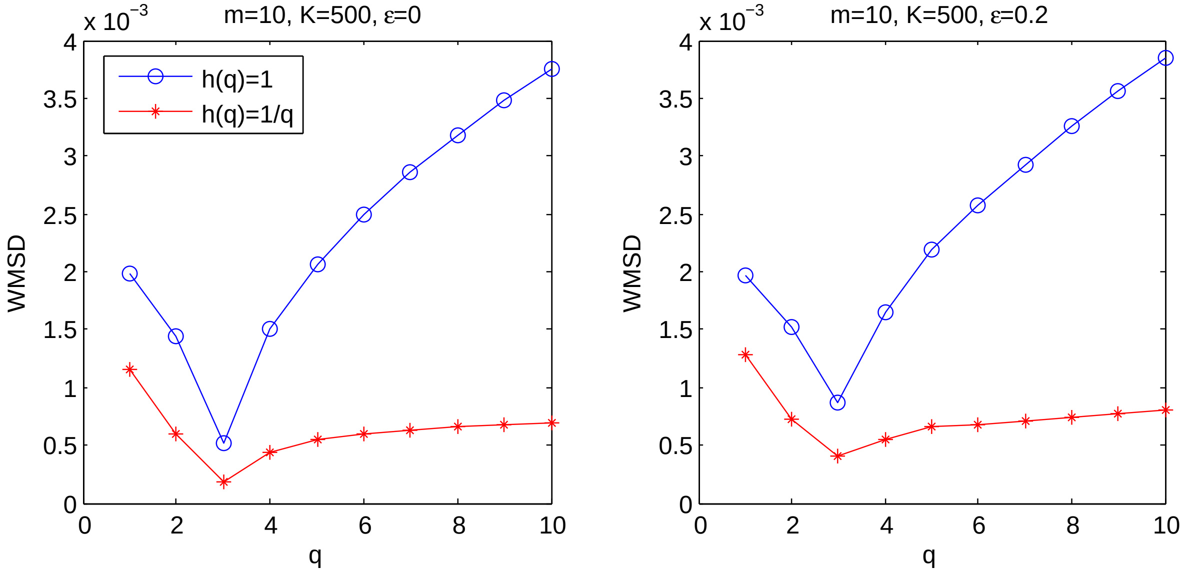

Figure 4.8: Select the number of components

- Projection Pursuit,
- Robust PCA based on MCD estimator.

For each method, the procedure is repeated 1000 times, and the mean square distance (MSD) between the true subspace and estimated subspace is evaluated by,

$$\text{MSD} = \frac{1}{1000}\sum_{i=1}^{1000} D^2(\mathcal{S}, \widehat{\mathcal{S}}^{(i)})\,.$$

Performance-Weighted Bagging PCA is calculated by algorithm 1 with $m = 10$, $K = 500$, $q = 3$. The weight w_k is calculated by the local rank correlation, with an exponential transformation $g(w_k) = \exp(5 \cdot w_k)$. The algorithm for calculating Projection Pursuit is provided in Hubert et al. [2002b] (RAPCA), and the MCD estimator is calculated by the FAST-MCD algorithm provided in Rousseeuw and Driessen [1999]. The simulation is conducted in MATLAB using the *LIBRA* library (Verboven and Hubert [2005]). In ROBPCA and MCD, the tuning parameter α, which specifies the fraction of outliers the algorithm should resist, is set to be $\alpha = 0.5$.

We report some results obtained from the following situations:

(i). $p = 5$, $\boldsymbol{\mu}_1 = c \cdot \mathbf{e}_5$, $\boldsymbol{\Sigma} = \text{diag}([8, 6, 4, 0.5, 0.1])$, $\epsilon = 0, \ldots, 0.45$, $c = 10,$ or 100. In this case, the true subspace $\mathcal{S} = \text{span}(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$.

(ii). $p = 50$, $\boldsymbol{\mu}_1 = c \cdot \mathbf{e}_{50}$, $\boldsymbol{\Sigma} = \text{diag}([25, 20, 18, 16, 15, 0.1, \ldots, 0.1])$, $\epsilon = 0, \ldots, 0.45$, $c = 10,$ or 100. In this case, the true subspace $\mathcal{S} = \text{span}(\mathbf{e}_1, \cdots, \mathbf{e}_5)$.

Case (i) considers the low-dimensional input data. The parameter c specifies how far the contaminated data are shifted from the clean data, $c = 10$ implies subtle outliers and $c = 100$ implies obvious outliers. The mean square distances are plotted in log scale in Figure 4.9.

From Figure 4.9, we can see that when the contamination level is low ($0 \leq \epsilon < 0.1$), Performance-Weighted Bagging PCA provides the best result no matter outliers are subtle or obvious. When the contamination level is high ($\epsilon > 0.4$), Performance-Weighted Bagging PCA also provides the best result in both situations. Notice that, although the resistance level for MCD and ROBPCA are set to be $\alpha = 0.5$, both methods are heavily influenced by outliers prior to the resistant level $\alpha = 0.5$. In this low-dimensional case, Performance-Weighted Bagging PCA yields a very robust subspace estimate against the presence of outliers.

Case (ii) considers the high-dimensional input data. The mean square distances are plotted in log scale in Figure 4.10. As can be seen in the left panel, Performance-Weighted Bagging PCA again produces a robust subspace estimate. When the contamination level

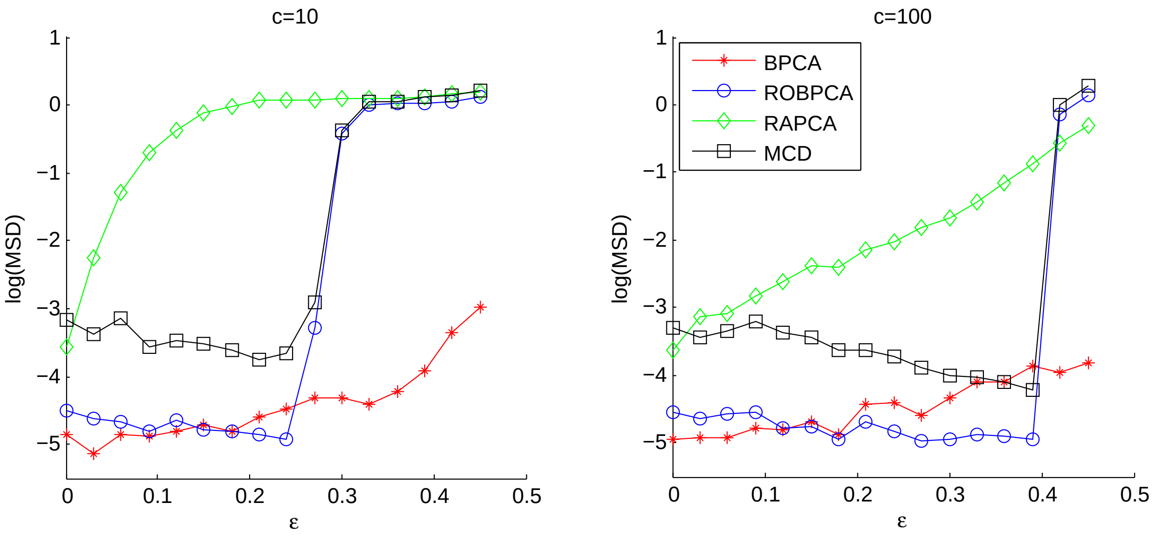

Figure 4.9: Simulation case (i).

is high, i.e. $\epsilon > 0.3$, Performance-Weighted Bagging PCA is the best among four robust methods. In the right panel, when $c = 100$, ROBPCA provides the best result, but Performance-Weighted Bagging PCA still outperforms RAPCA and robust PCA based on MCD estimator.

4.5 Background modeling from surveillance video

Analysis of video data is an active research field. In this section, we apply Performance-Weighted Bagging PCA on the background modeling problem (Bouwmans et al. [2014]) in surveillance video data analysis. In a surveillance video, every scene usually consists of a relatively static background, and some moving objects. A basic task is to separate the background and foreground objects. This problem is appealing in computer vision, and it is usually referred as *background modeling* or *foreground detection*. Background

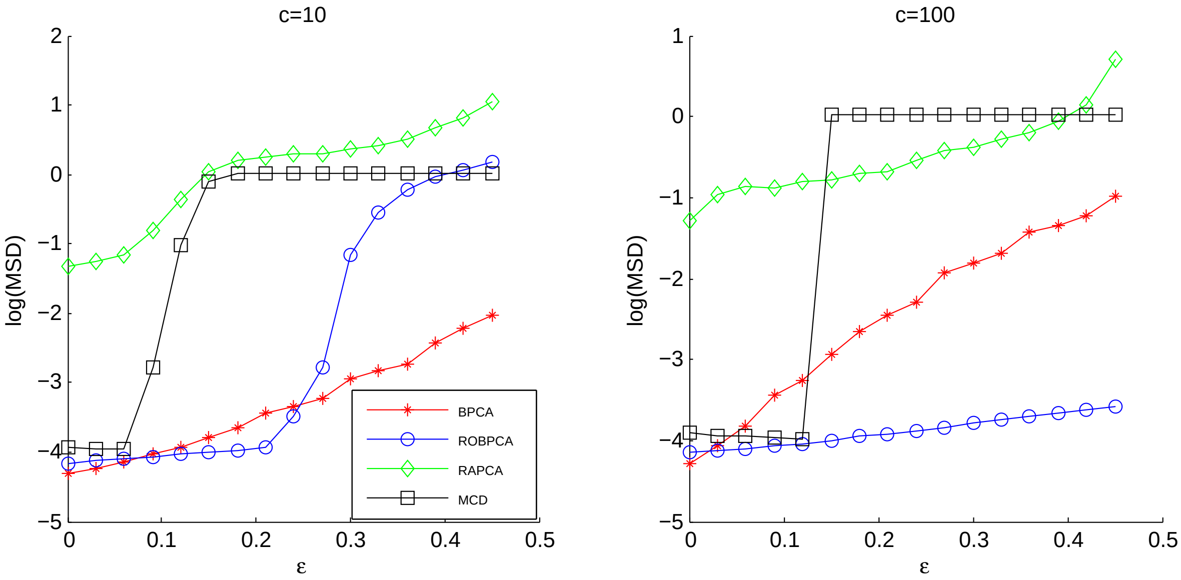

Figure 4.10: Simulation case (ii).

modeling can be challenging in practice due to changes of background, for example changes in illumination condition, presence of shadows, or due to a poor quality video source. In such situations, one way to tackle the problem is via subspace learning models (Bouwmans [2009]).

We stack the sequence of image frames from a video as column vectors $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, where n is the number of frames and the dimensionality p of $\mathbf{x}_i$ is the resolution of each frame. The background can be modeled by a low-dimensional subspace $\mathcal{S}_\mathbf{P} \subset \mathbb{R}^p$. For the i-th frame, the background are represented by the mean image $\mathbf{m}$ and projection matrix $\mathbf{P}$ associated with the subspace $\mathcal{S}_\mathbf{P}$, and the foreground objects is represented by the difference between the original image and its reconstruction, i.e.

$$\text{mean image: } \mathbf{m} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i\,,$$

$$\text{background: } \widehat{\mathbf{x}}_j = \mathbf{m} + \mathbf{P}\cdot(\mathbf{x}_j - \mathbf{m})\,,$$

$$\text{foreground: } \widehat{\mathbf{r}}_j = \mathbf{x}_j - \widehat{\mathbf{x}}_j\,.$$

Here we will apply Performance-Weighted Bagging PCA to obtain a subspace $\mathcal{S}_{\mathbf{P}}$ and investigate its performance. Three surveillance video datasets are considered here as illustration, and all calculations are conducted in MATLAB. The first dataset is introduced in Li et al. [2004]. It is a sequence of $n = 1500$ grayscale image frames of a escalator recorded by a CCTV surveillance system, and each frame has resolution $p = 130 \times 160$. The moving objects in the scene are walking people and escalator steps. This dataset is challenging for three reasons. First, there are many more moving objects in the scene, including three escalators and busy flow of human crowds. Another challenge is due to the significant change in the background lighting conditions. In addition, the video is noisy because of the old video recording device used. We first apply Performance-Weighted Bagging PCA in algorithm 1, with $m = 50$, $K = 300$, and $q = 3$. The result is shown is Figure 4.11. Three frames are randomly chosen from the original video and displayed in the left column, and the middle and right columns show the segmentation of the background and the foreground objects.

Secondly, we do not specify the number of components q, and apply Performance-Weighted Bagging PCA in algorithm 2, with $m = 50$, $K = 300$, and the stopping rule is chosen to be

$$q_{_k} = q(\mathbf{X}_k) = \min\left\{ q : \frac{\sum_{j=1}^{q} \lambda_{_{(k),j}}}{\sum_{j=1}^{p} \lambda_{_{(k),j}}} > 0.75 \right\}\,,$$

where $\lambda_{_{(k),j}}$ is the j-th largest eigenvalue of the covariance matrix of $\mathbf{X}_k$. We also apply principal component pursuit (Candès et al. [2011]), which is a state-of-the-art technique, for

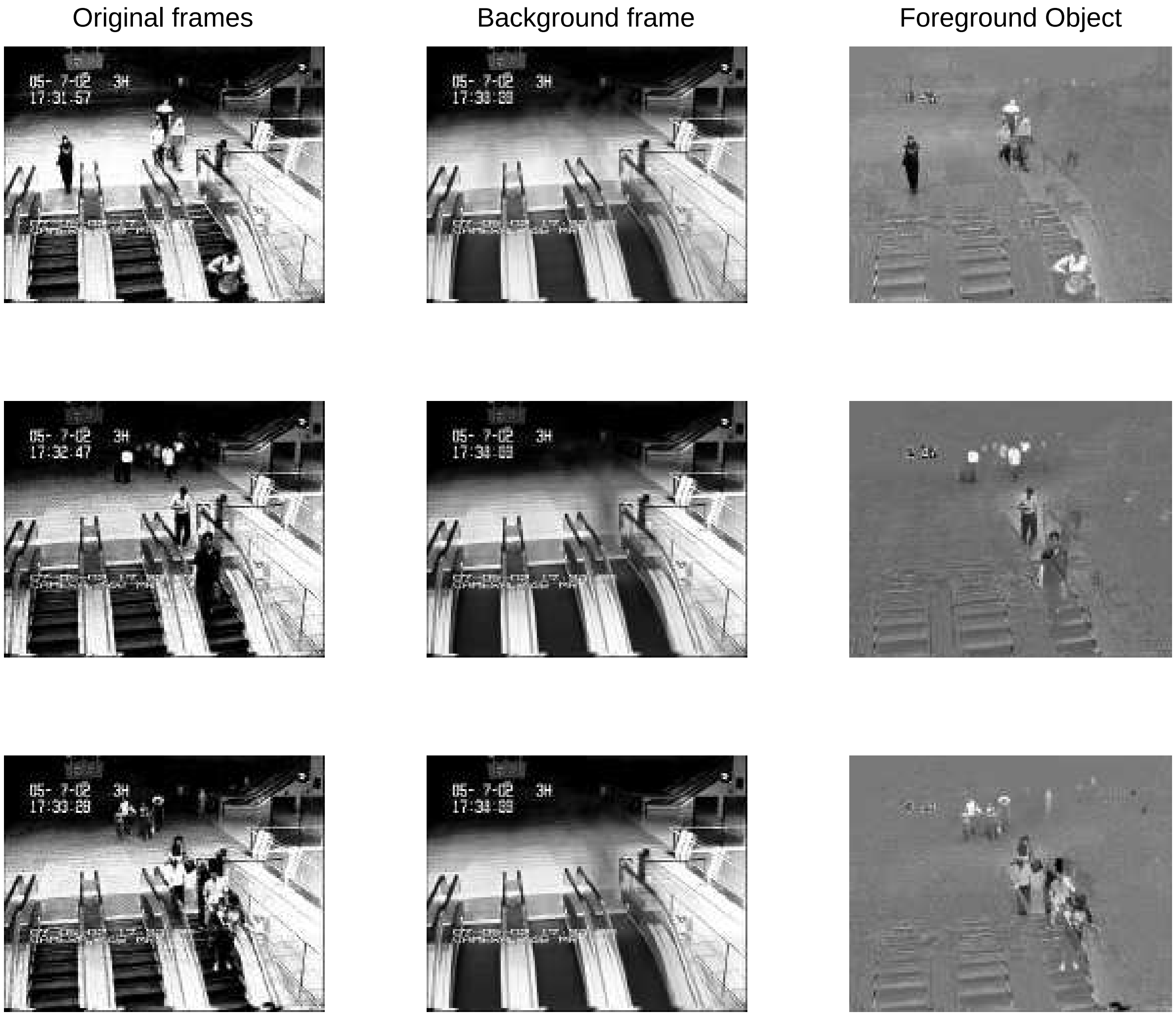

Figure 4.11: Background modeling in escalator surveillance video data by Performance-Weighted Bagging PCA (algorithm 1).

comparison. We use the MATLAB code package downloaded from `http://perception.csl.illinois.edu/matrix-rank/sample_code.html`. The result is shown in Figure 4.12.

Image frames from the original video are displayed in column (a), column (b) and

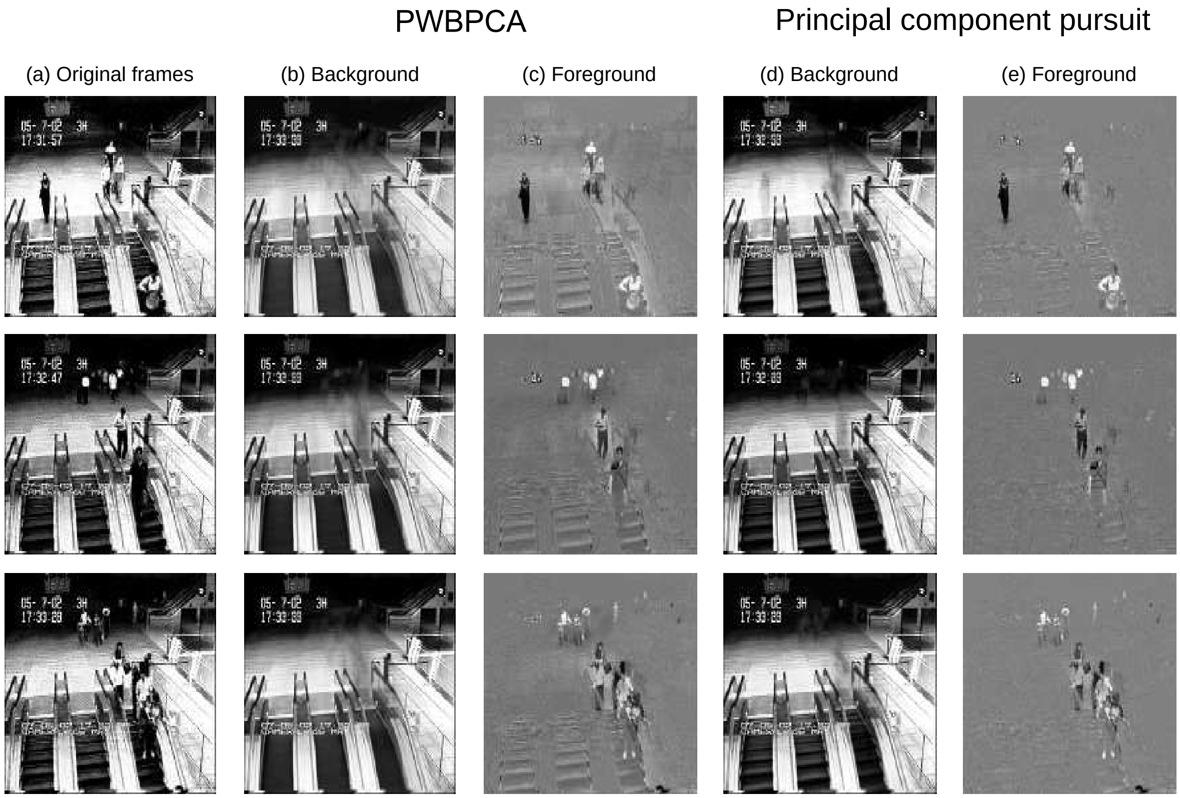

Figure 4.12: Background modeling in escalator surveillance video data by Performance-Weighted Bagging PCA (algorithm 2) and principal component pursuit.

(c) show the segmentation of the background and foreground obtained from Performance-Weighted Bagging PCA, whereas column (d) and (e) show the segmentation of the background and foreground obtained from principal component pursuit. As can be seen from Figure 4.12 column (b) and (d), the main difference between Performance-Weighted Bagging PCA segmentation and principal component pursuit segmentation is that escalator steps are treated as moving objects in Performance-Weighted Bagging PCA while principal component pursuit identifies them as background. In general, Performance-Weighted

Bagging PCA provides a visually satisfactory result in spite of the illumination changes in the background and the poor quality of the video.

The second dataset is from *CVLAB-EPFL* (`http://cvlab.epfl.ch/data`). It is a sequence of $n = 1000$ grayscale image frames filmed by a static camera at a training session of a local basketball team, and each frame has resolution $p = 144 \times 180$. The moving objects are basketball players and the basketball. The goal is to track players by separating them from the background basketball court. The difficulty in dealing with this dataset is due to frequent and complicated movements of players. We again apply Performance-Weighted Bagging PCA and principal component pursuit with the same parameter setting in the escalator surveillance data example. The result is shown in Figure 4.13, and Figure 4.14.

As can be seen from Figure 4.14, all players in the scene are successfully captured by both Performance-Weighted Bagging PCA and principal component pursuit. However, both methods treat some of the player motion as background, causing the presence of "ghost" in some background frames. In these two challenging video datasets, Performance-Weighted Bagging PCA gives a competitive result, compared to the state-of-the-art technique.

The third example illustrates the performance of the proposed method on a contaminated dataset. The dataset is also introduced in Li et al. [2004]. It is a surveillance video taken in a lobby with resolution $p = 96 \times 120$. The scene contains several moving people, as well as several drastic illumination changes. We select a sequence of $n = 230$ grayscale images, and then we contaminated the dataset by adding three random noise images. As shown in Figure 4.15, the upper row (a) shows three frames from the original video, and lower row (b) shows three noisy images. Note that pixels of three added images have magnitude approximately 4 times more than other clean images.

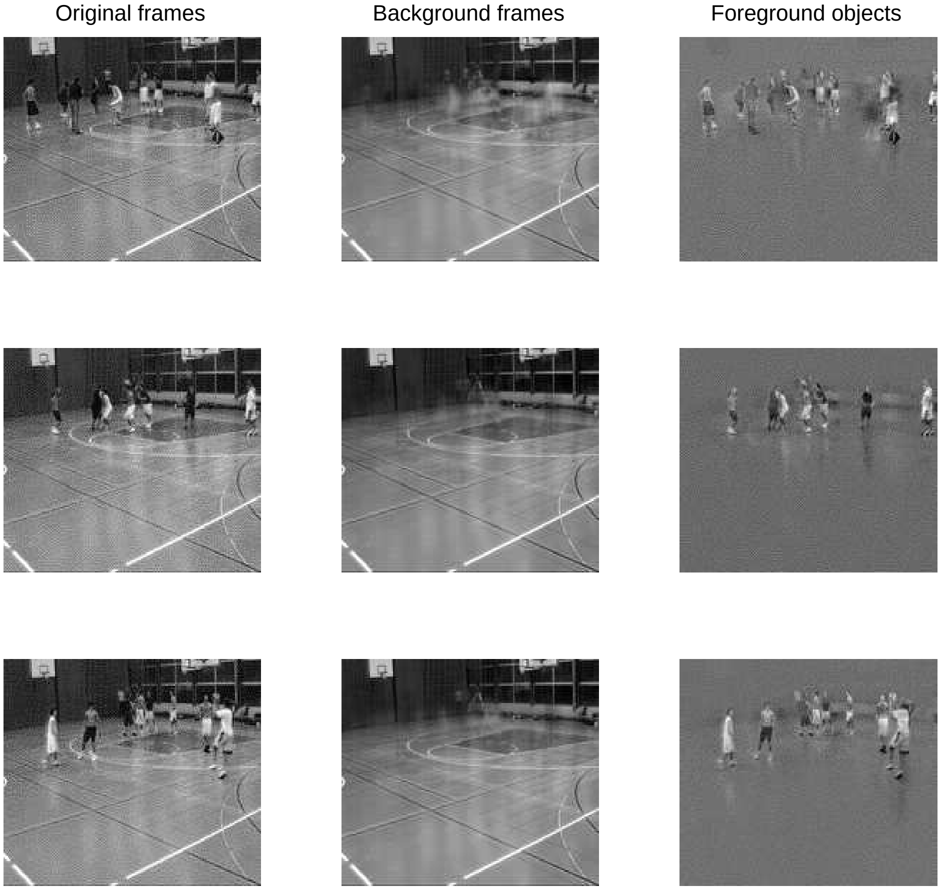

Figure 4.13: Player detection in basketball video data by Performance-Weighted Bagging PCA (algorithm 1).

The goal is to identify the moving people from each frame as foreground objects and illumination changes as a part of the background. The main challenge comes from three added noisy images. We apply four different PCA variants for comparison. They are traditional PCA, principal component pursuit, ROBPCA, and proposed Performance-Weighted Bagging PCA. The number of components for PCA, ROBPCA, and Performance-Weighted Bagging PCA is chosen to be $q = 3$, and the resistant level for ROBPCA is chosen to be

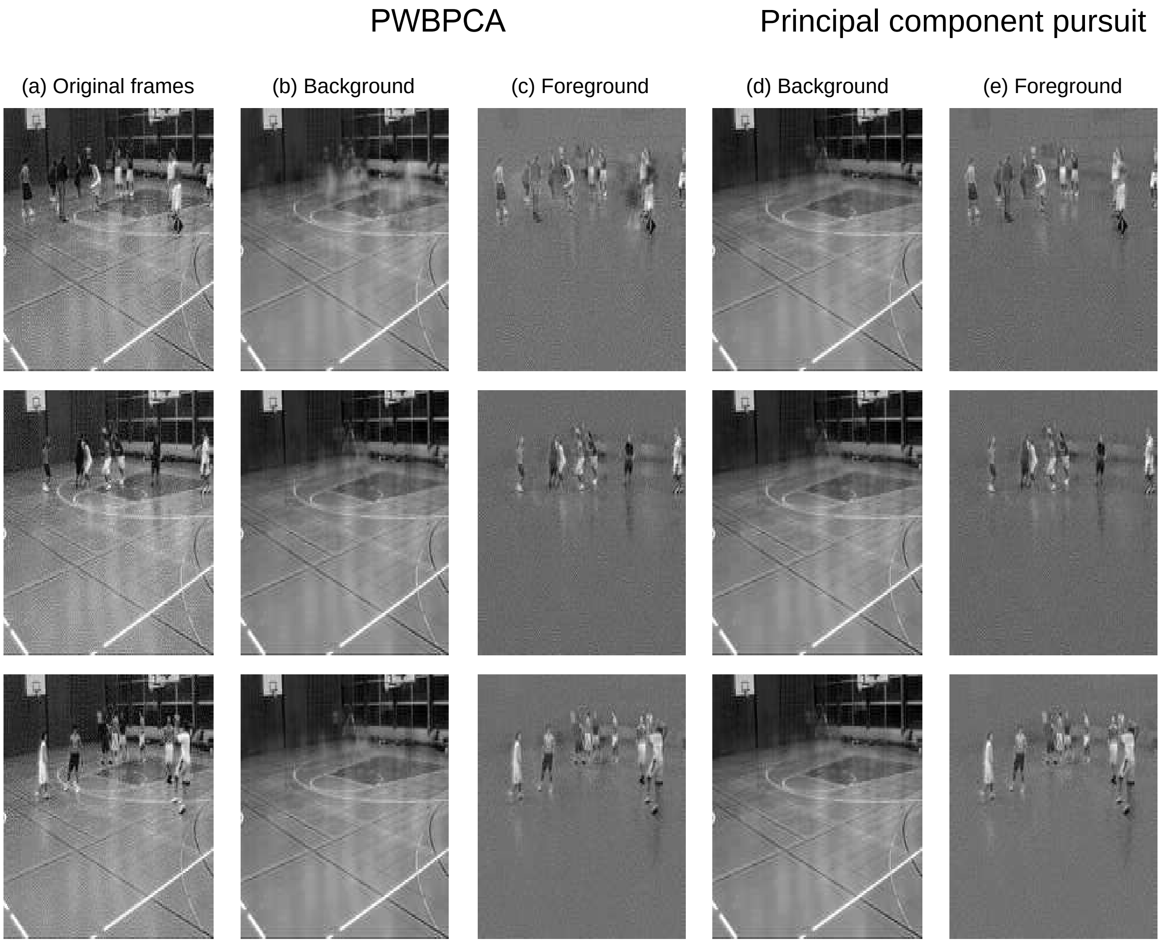

Figure 4.14: Player detection in basketball video data by Performance-Weighted Bagging PCA (algorithm 2) and principal component pursuit.

$\alpha = 0.6$. First we apply these four methods on the clean dataset, and the result is shown in Figure 4.16. Three frames from the original video are randomly selected, and displayed in column (a). Column (b)-(e) display the foreground detection from four methods, respectively. As can be seen, PCA, principal component pursuit, and Performance-Weighted Bagging PCA all yield reasonably good results, while ROBPCA, a traditional robust method,

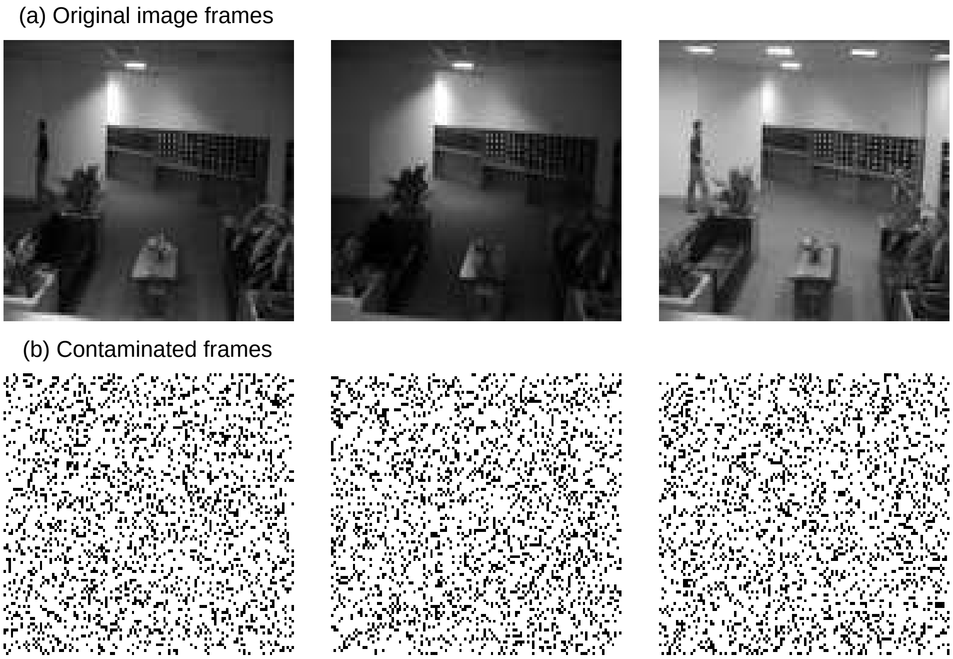

Figure 4.15: Three frames from the original video (upper row), and three random noise images (lower row).

fails to separate the moving people from the background.

Then we apply these four methods on the contaminated dataset, and the result is shown in Figure 4.17. As can be seen, when outlying images are included in the dataset, the subspace obtained from PCA can no longer capture the illumination changes. Compared to PCA, ROBPCA also fails to provide a reasonable result, while principal component pursuit (PCP) successfully recovers the moving objects from the contaminated data. Our proposed Performance-Weighted Bagging PCA works reasonably well. Although a part of illumination changes is mistakenly considered as foreground objects, overall the segmentation is robust against outliers, compared to PCA and ROBPCA.

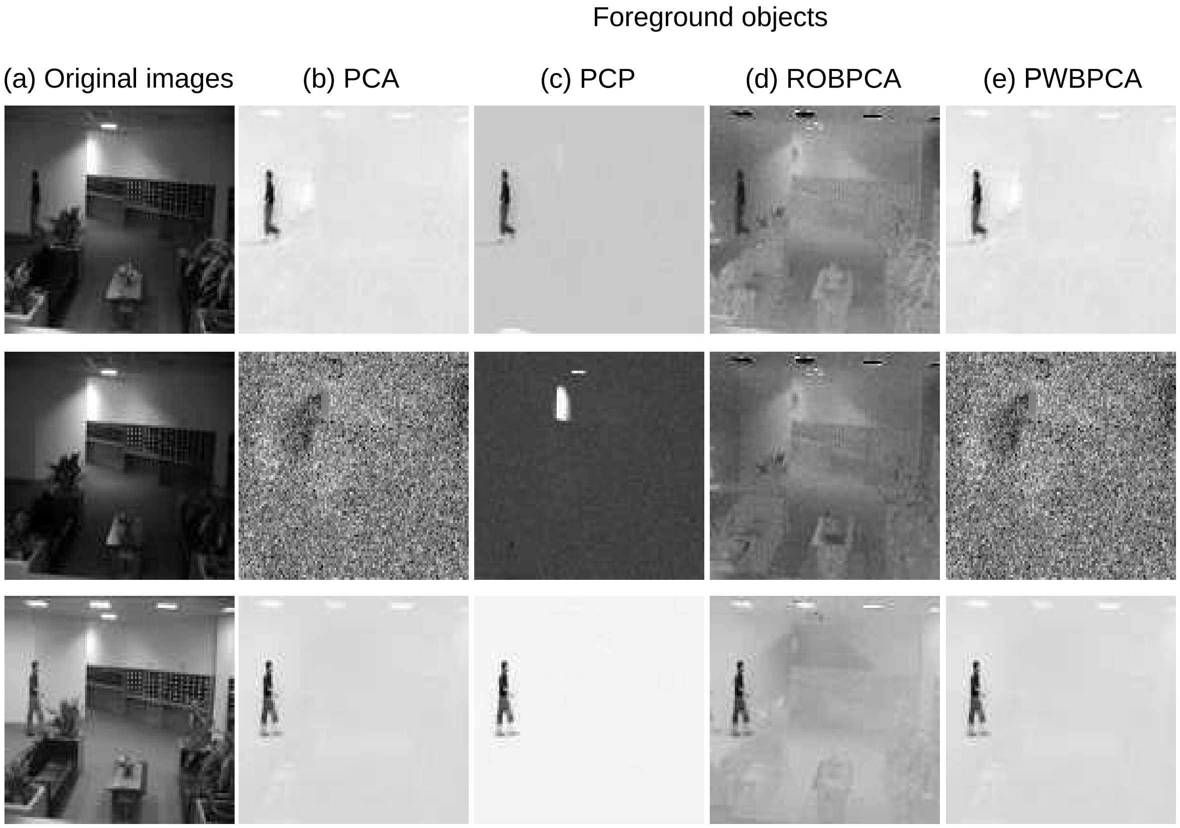

Figure 4.16: Foreground objects detection with clean data.

4.6 Discussion and future Work

In this chapter we developed a new robust PCA approach from the viewpoint of model averaging. The proposed method is robust and computationally convenient. It yields competitive results in numerical examples compared to some traditional robust PCA methods, and it also shows the applicability in the analysis of video data. There are still several interesting problems in Performance-Weighted Bagging PCA which can be future research topics.

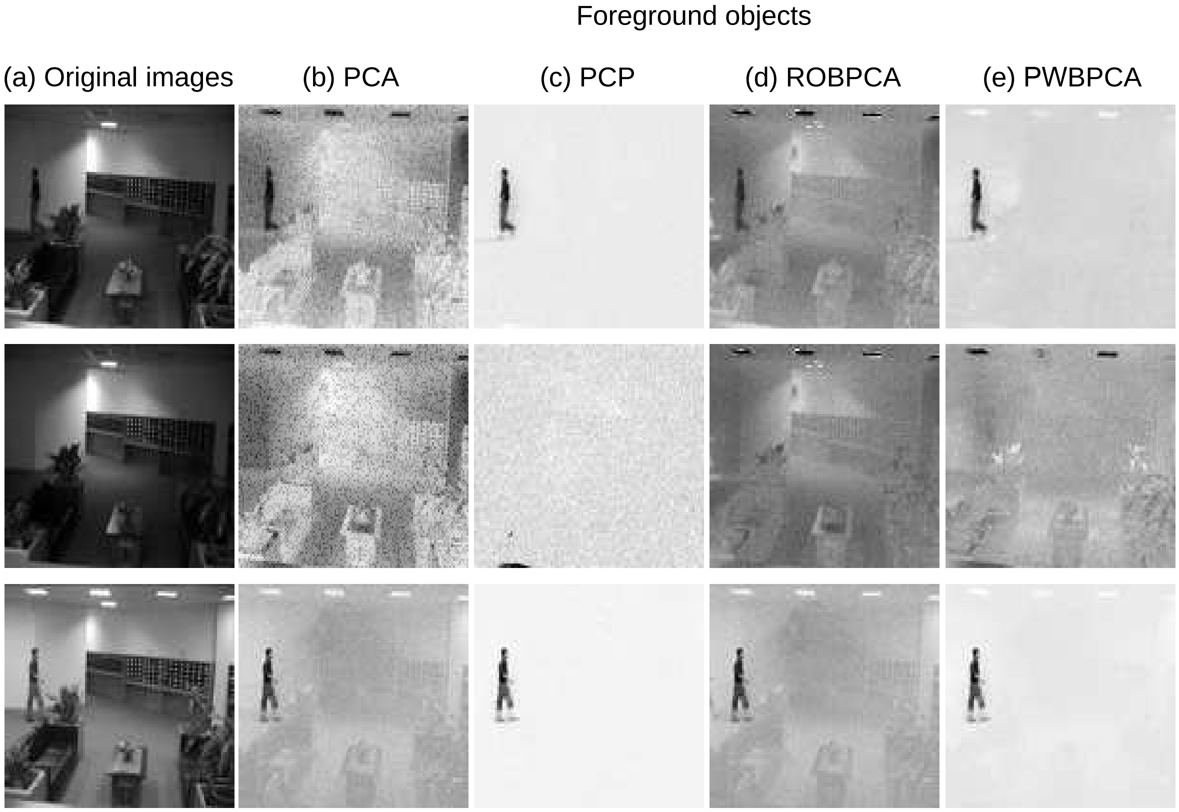

Figure 4.17: Foreground objects detection with contaminated data.

An immediate goal is to extend Performance-Weighted Bagging PCA to handle data with a nonlinear structure via Kernel PCA framework. Kernel PCA first transforms the input data into a feature space $\mathcal{H}$ by a feature map Φ, and traditional PCA is then applied in $\mathcal{H}$ to find a subspace for projection. It is difficult to directly generalize the idea of averaging subsample subspaces in the feature space due to the implicitness of Φ. An alternative way of performing model averaging in Kernel PCA is to average over a set of candidate kernel matrices. Some methods contain tuning parameters in constructing kernel matrices, for example the neighborhood size K in LLE and ISOMAP, and parameter σ

in Gaussian kernel PCA. Instead of choosing a optimal value for the tuning parameter, we can use the idea of bagging. First construct kernel matrix for a set of different values of tuning parameter, and then assign a proper weight for each kernel matrix based on the performance of the corresponding output configuration. The weighted average kernel matrix will produce an outcome that is robust against the tuning parameter.

A second topic to be explored is the weight function w_k. In general, the various choices in the weighting scheme introduce more flexibility and potential in this framework. A further study is needed to better understand the behavior of weighting function, and to choose the optimal family of weighting functions. Also, as we have seen in the example in Section 4.2.3, the transformation plays another important role. A proper increasing transformation can magnify the effect of "good" subsample subspaces, and improve the robustness of Performance-Weighted Bagging PCA. However, a sharply increasing transformation might force a large proportion of subsample subspaces to have nearly 0 weights, causing the loss of efficiency. It seems like the transformation $g(\cdot)$ and the subsample size m together controls the trade-off between the efficiency and robustness of Performance-Weighted Bagging PCA. How can we find the optimal subsample size and the optimal transformation to balance between the efficiency and the robustness is an important issue in the future research.

In addition, it is also of our interest to derive some theoretical properties of Performance-Weighted Bagging PCA. For example under some distributional assumptions, we would like to investigate the asymptotic property of the distance $D(\widehat{\mathcal{S}}_{Bag}, \mathcal{S})$ between Performance-Weighted Bagging PCA subspace $\widehat{\mathcal{S}}_{Bag}$ and underlying subspace $\mathcal{S}$.

References

Ammann, L. P. (1993). Robust singular value decompositions: A new approach to projection pursuit. *Journal of the American Statistical Association*, 88(422):505–514.

Andrews, D. F., Bickel, P. J., Hampel, F. R., Huber, P. J., Rogers, W. H., and Tukey, J. W. (1972). *Robust Estimates of Location*. Princeton University Press, Princeton, NJ.

Armstrong, M. A. (1979). *Basic topology*. McGraw-Hill, London.

Barnett, V. and Lewis, T. (1984). *Outliers in statistical data*. Wiley, New York, NY, 2 edition.

Bauer, H. U. and Pawelzik, K. R. (1992). Quantifying the neighborhood preservation of self-organizing maps. *IEEE Transactions on Neural Networks*, 3:570–579.

Becker, S., Candes, E., and Grant, M. (2011). Tfocs: Flexible first-order methods for rank minimization. In *Low-rank Matrix Optimization Symposium, SIAM Conference on Optimization*.

Beckman, F. E. and Cook, R. D. (1983). Outlier..........s. *Technometrics*, 25:119–149.

Belkin, M. and Niyogi, P. (2001). Laplacian eigenmaps and spectral techniques for embedding and clustering. *Advances in neural information processing systems*, 14:585–591.

Belkin, M. and Niyogi, P. (2003). Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15:1373–1396.

Bellman, R. E. (1961). *Adaptive control processes: a guided tour.* Princeton University Press, Princeton, NJ.

Bénasséni, J. (1990). Sensitivity coefficients for the subspaces spanned by principal components. *Communications in Statistics-Theory and Methods*, 19(6):2021–2034.

Bernholt, T. and Fischer, P. (2004). The complexity of computing the mcd-estimator. *Theoretical Computer Science*, 326(1):383–398.

Bernstein, M., de Silva, V., Langford, J. C., and Tenenbaum, J. B. (2002). Graph approximations to geodesics on embedded manifolds. Technical report, Stanford University.

Boente, G., Fraiman, R., and Yohai, V. J. (1987). Qualitative robustness of rank tests. *The Annals of Statistics*, 15:1293–1312.

Boente, G., Pires, A. M., and Rodrigues, I. (2002). Influence functions and outlier detection under the common principal components model: A robust approach. *Biometrika*, 89:861–875.

Bouwmans, T. (2009). Subspace learning for background modeling: A survey. *Recent Patents on Computer Science*, 2(3):223–234.

Bouwmans, T., Porikli, F., Höferlin, B., and Vacavant, A. (2014). *Background Modeling and Foreground Detection for Video Surveillance.* CRC Press.

Box, G. E. P. (1953). Non-nomality and tests on variance. *Biometrika*, 40:318–335.

Breiman, L. (1996). Bagging predictors. *Machine learning*, 24(2):123–140.

Bruske, J. and Sommer, G. (1998). Intrinsic dimensionality estimation with optimally topology preserving maps. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(5):572–575.

Büchlmann, P. and Yu, B. (2002). Analyzing bagging. *Annals of Statistics*, pages 927–961.

Buja, A. and Stuetzle, W. (2006). Observations on bagging. *Statistica Sinica*, 16(2):323.

Calder, P. (1986). *Influence functions in multivariate analysis.* PhD thesis, University of Kent.

Camastra, F. and Vinciarelli, A. (2002). Estimating the intrinsic dimension of data with a fractal-based method. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(10):1404–1407.

Campbell, N. A. (1980). Robust procedures in multivariate analysis i: Robust covariance estimation. *Journal of the Royal Statistical Society. Series C*, 29:231–237.

Candès, E. J., Li, X., Ma, Y., and Wright, J. (2011). Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3):11.

Candès, E. J. and Recht, B. (2009). Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717–772.

Carreira-Perpinan, M. A. (1997). A review of dimension reduction techniques. Technical report CS-96-09, Department of Computer Science, University of Sheffield.

Castaão-Tostado, E. and Tanaka, Y. (1990). Some comments on escoufier's rv-coefficient as a sensitivity measure in principal component analysis. *Communications in Statistics-Theory and Methods*, 19(12):4619–4626.

Chandrasekaran, V., Sanghavi, S., Parrilo, P. A., and Willsky, A. S. (2011). Rank-sparsity incoherence for matrix decomposition. *SIAM Journal on Optimization*, 21(2):572–596.

Chang, H. and Yeung, D. Y. (2006). Robust locally linear embedding. pattern recognition. *Pattern recognition*, 39:1053–1065.

Chen, L. and Buja, A. (2009). Local multidimensional scaling for nonlinear dimension reduction, graph drawing, and proximity analysis. *Journal of the American Statistical Association*, 104:209–219.

Chenouri, S., Liang, J., and Small, C. G. (2015). Robust dimension reduction. *Wiley Interdisciplinary Reviews: Computational Statistics*, 7(1):63–69.

Choi, H. and Choi, S. (2007). Robust kernel isomap. *Pattern recognition*, 39:853–862.

Cook, R. D. and Weisberg, S. (1980). Characterizations of an empirical influence function for detecting influential cases in regression. *Technometrics*, 22:495–508.

Cortes, C. and Vapnik, V. (1995). Support vector networks. *Machine learning*, 20:273–297.

Cox, T. F. and Cox, M. A. A. (1994). *Multidimensional scaling*. Chapman & Hall, London.

Critchley, F. (1985). Influence in principal components analysis. *Biometrika*, 72:627–636.

Crone, L. and Crosby, D. (1995). Statistical applications of a metric on subspaces to satellite meteorology. *Technometrics*, 37(3):324–328.

Croux, C. (1999). Influence function and efficiency of the minimum covariance determinant scatter matrix estimator. *Journal of Multivariate Analysis*, 71:161–190.

Croux, C. and Haesbroeck, G. (1999). Influence function and efficiency of the minimum covariance determinant scatter matrix estimator. *Journal of Multivariate Analysis*, 71(2):161–190.

Croux, C. and Haesbroeck, G. (2000). Principle components analysis based on robust estimators of the covariance or correlation matrix: Influence functions and efficiencies. *Biometrika*, 87:603–618.

Croux, C. and Ruiz-Gazen, A. (1996). A fast algorithm for robust principal components based on projection pursuit. In *COMPSTAT 1996, Proceedings in Computational Statistics*, pages 211–217, Heidelberg. Physica-Verlag.

Cuevas, A. (1988). Qualitative robustness in abstract inference. *Journal of Statistical Planning and Inference*, 18:277–289.

Davies, L. (1987). Asymptotic behavior of s-estimators of multivariate location and dispersion matrices. *The Annals of Statistics*, 15:1269–1292.

Davies, P. (1993). Aspects of robust linear regression. *The Annals of Statistics*, 21:1843–1899.

Davies, P. and Gather, U. (2005). Breakdown and groups. *The Annals of Statistics*, 33:977–1035.

Davies, P. and Gather, U. (2007). The breakdown point–examples and counterexamples. *REVSTAT Statistical Journal*, 5:1–17.

Debruyne, M., Hubert, M., and Van Horebeek, J. (2010). Detecting influential observations in kernel pca. *Computational Statistics & Data Analysis*, 54(12):3007–3019.

Devlin, S. J., Gnanadesikan, R., and Kettenring, J. R. (1981). Robust estimation of dispersion matrices and principal components. *Journal of the American Statistical Association*, 76(374):354–362.

Dietterich, T. G. (2000). Ensemble methods in machine learning. In *Multiple classifier systems*, pages 1–15. Springer.

Dijkstra, E. W. (1959). A note on two problems in connection with graphs. *Numerische Mathematik*, 1:269–271.

Dixon, W. J. (1953). Processing data for outliers. *Biometrics*, 9:74–89.

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

Donoho, D. L. (1982). *Breakdown properties of multivariate location estimators.* PhD thesis, Dept. Statistics, Harvard University.

Donoho, D. L. (2000). High-dimensional data analysis: the curse and blessings of dimensionality. In *Aide-Memoire of the lecture in AMS conference "Math challenges of 21st Centrury"*.

Donoho, D. L. and Gasko, M. (1992). Breakdown properties of location estimates based on halfspace depth and projected outlyingness. *The Annals of Statistics*, 20:1803–1827.

Donoho, D. L. and Grimes, C. (2002a). Local isomap perfectly recovers the underlying parametrization for familites of occluded/lacunary images. Technical report, Stanford University.

Donoho, D. L. and Grimes, C. (2002b). When does isomap recover the natural parameterization of families of articulated images. Technical report, Stanford University.

Donoho, D. L. and Huber, P. J. (1983). The notion of breakdown point. In *A Festschrift for Erich L. Lehmann*, pages 157–184, Belmont, CA. Wadsworth, Inc.

Du, C., Sun, J., Zhou, S., and Zhao, J. (2013). An outlier detection method for robust manifold learning. In *Proceedings of The Eighth International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA)*, pages 353–360. Springer Berlin Heidelberg.

Efron, B. and Tibshirani, R. J. (1994). *An introduction to the bootstrap.* CRC press.

Ellis, S. P. and Morgenthaler, S. (1992). Leverage and breakdown in l_1 regression. *Journal of the American Statistical Association*, 87:143–148.

Escoufier, Y. (1973). Le traitement des variables vectorielles. *Biometrics*, pages 751–760.

Ferré, L. (1995). Selection of components in principal component analysis: a comparison of methods. *Computational Statistics & Data Analysis*, 19(6):669–682.

Fieller, N. R. J. (1976). *Some problems related to the Rejection of Outlying observations.* PhD thesis, University of Hull.

Floyd, R. W. (1962). Algorithm 97: shortest path. *Communications of the ACM*, 5:345.

Fodor, I. K. (2002). A survey of dimension reduction techniques. *Technical report UCRL-ID-148494, LLNL.*

France, S. and Carroll, D. (2007). Development of an agreement metric based upon the rand index for the evaluation of dimensionality reduction techniques, with applications to mapping customer data. In *Machine learning and data mining in pattern recognition*, pages 499–517. Springer.

Friedman, J., Hastie, T., and Tibshirani, R. (2009). *Elements of Statistical Learning: Prediction, Inference and Data Mining*. Springer, New York, NY.

Fukunaga, K. and Olsen, D. R. (1971). An algorithm for finding intrinsic dimensionality of data. *Computers, IEEE Transactions on*, 100:176–183.

Gabrys, B., Baruque, B., and Corchado, E. (2006). Outlier resistant pca ensembles. In *Knowledge-Based Intelligent Information and Engineering Systems*, pages 432–440. Springer.

Gastwirth, J. and Rubin, H. (1971). Effect of dependence on the level of some one-sample tests. *Journal of the American Statistical Association*, 66:816–820.

Gather, U. and Becker, C. (1997). Outlier identification and robust methods. In *Handbook of Statistics*, 15, chapter 6, pages 123–144. Elsevier Science, North-Holland. Robust Inference.

Genton, M. G. (2003). Breakdown-point for spatially and temporally correlated observations. In *Developments in Robust Statistics, International conference on Robust Statistics 2001*, pages 148–159. Physica, Heidelberg.

Genton, M. G. and Lucas, A. (2003). Comprehensive definitions of breakdown points for independent and dependent observations. *Journal of the Royal Statistical Society, B*, 65:81–94.

Gnandesikan, R. and Wilk, M. (1969). Data analytic methods in multivariate statistical analysis. In Krishnaiah, P. R., editor, *Multivariate Analysis II.* Academic Press, New York, NY.

Goldberg, Y. and Ritov, Y. (2009). Local procrustes for manifold embedding: a measure of embedding quality and embedding algorithms. *Machine learning*, 77(1):1–25.

Goldberg, Y., Zakai, A., Kushnir, D., and Ritov, Y. (2008). Manifold learning: the price of normalization. *Journal of Machine Learning Research*, 9:1909–1939.

Gordaliza, A. (1991). On the breakdown point of multivariate location estimators based on trimming procedures. *Statistics & Probability Letters*, 11:387–394.

Grassberger, P. and Procaccia, I. (2004). Measuring the strangeness of strange attractors. In *The Theory of Chaotic Attractors*, pages 170–189. Springer.

Grubbs, F. E. (1969). Procedures for detecting outlying observations in samples. *Technometrics*, 11:1–21.

Ham, J., Lee, D. D., Mika, S., and Schölkopf, B. (2004). A kernel view of the dimensionality reduction of manifolds. In *Proceedings of the twenty-first international conference on Machine learning*, page 47. ACM.

Hampel, F. R. (1968). *Contributions to the Theory of Robust Estimation.* PhD thesis, University of California, Berkeley.

Hampel, F. R. (1971). A general qualitative definition of robustness. *Annals of Mathematical Statistics*, 42:1887–1896.

Hampel, F. R. (1974). The influence curve and its role in robust estimation. *Journal of the American Statistical Association*, 69:383–393.

Hampel, F. R. (1975). Beyond location parameters: Robust concepts and methods (with discussion). *Bulletin of the International Statistical Institute*, 46:375–391.

Hampel, F. R. (1985). The breakdown points of the mean combined with some rejection rules. *Technometrics*, 27:95–107.

Hampel, F. R. (2000). Robust inference. In *Encyclopedia of Environmentrics*. Research Report 93, Seminar für Statistik, ETH Zürich.

Hampel, F. R. (2001). Robust statistics: A brief introduction and overview. Technical report, Swiss Federal Institute of Technology (ETH), Zürich, Zürich. Invited talk in the Symposium "Robust Statistics and Fuzzy Techniques in Geodesy and GIS".

Hampel, F. R. (2005). Discussion of breakdown and groups by P. L. Davies and U. Gather. *The Annals of Statistics*, 33:993–998.

Hampel, F. R., Ronchetti, E. M., Rousseeuw, P. J., and Stahel, W. A. (1986). *Robust Statistics: The Approach Based on Influence Functions*. Wiley, New York.

Hastie, T. and Steutzle, W. (1989). Principal curves. *Journal of the American Statistical Association*, 84:502–516.

Hawkins, D. M. (1980). *Identification of Outliers*. Chapman & Hall, London.

He, X. (1991). A local breakdown property of robust tests in linear regression. *Journal of Multivariate Analysis*, 38:294–305.

He, X. and Fung, W. K. (2000). High breakdown estimation for multiple populations with applications to discriminant analysis. *Journal of Multivariate Analysis*, 72:151–162.

He, X. and Portnoy, S. L. (2000). A robust journey in the new millennium. *Journal of the American Statistical Association*, 95:1331–1335.

He, X. and Simpson, D. G. (1992). Robust direction estimation. *The Annals of Statistics*, 20:351–369.

He, X. and Simpson, D. G. (1993). Lower bounds for contamination bias: Globally minimax versus locally linear estimation. *The Annals of Statistics*, 21:314–337.

He, X., Simpson, D. G., and Portnoy, S. L. (1990). Breakdown robustness of tests. *Journal of the American Statistical Association*, 95:1331–1335.

Higuchi, I. and Eguchi, S. (1998). The influence function of principal component analysis by self-organizing rule. *Neural Computation*, 10:1435–1444.

Higuchi, I. and Eguchi, S. (2004). Robust principal component analysis with adaptive selection for tuning parameters. *Journal of Machine Learning Research*, 5:453–471.

Hoaglin, D. C., Mosteller, F., and Tukey, J. W. (1983). *Understanding Robust and Exploratory Data Analysis*. Wiley, New York, NY.

Hodge, V. and Austin, J. (2004). A survey of outlier detection methodologies. *Artificial Intelligence Review*, 22:85–126.

Hodges, J. L. J. (1967). Efficiency in normal samples and tolerance of extreme values for some estimates of location. In *Proc. Fifth Berkeley symp. Math. Statist. Prob.*, volume 1, pages 163–186. University of California Press.

Hsu, D., Kakade, S. M., and Zhang, T. (2011). Robust matrix decomposition with sparse corruptions. *Information Theory, IEEE Transactions on*, 57(11):7221–7234.

Huang, S. Y., Yeh, Y. R., and Eguchi, S. (2009). Robust kernel principal component analysis. In *Neural computation*, volume 21, pages 3179–3213.

Huber, P. J. (1964). Robust estimation of a location parameter. *Annals of Mathematical Statistics*, 35:73–101.

Huber, P. J. (1965). A robust version of the probability ratio test. *Annals of Mathematical Statistics*, 36:1753–1758.

Huber, P. J. (1972). Robust statistics: A review. *Annals of Mathematical Statistics*, 43:1041–1067.

Huber, P. J. (1981). *Robust Statistics*. Wiley, New York, NY.

Huber, P. J. and Ronchetti, E. M. (2011). *Robust Statistics*. Wiley, New York, NY, 2 edition.

Hubert, M., Rousseeuw, P. J., and Branden, K. V. (2005). Robpca: a new approach to robust principal component analysis. *Technometrics*, 47:64–79.

Hubert, M., Rousseeuw, P. J., and Verboven, S. (2002a). A fast method for robust principal components with applications to chemometrics. *Chemometrics and Intelligent Laboratory Systems*, 60(1):101–111.

Hubert, M., Rousseeuw, P. J., and Verboven, S. (2002b). A fast method for robust principal components with applications to chemometrics. *Chemometrics and Intelligent Laboratory Systems*, 60:101–111.

Huo, X. and Smith, A. K. (2006). Performance analysis of a manifold learning algorithm in dimension reduction. Technical report, Georgia Institute of Technology.

Jackson, J. E. (2005). *A user's guide to principal components*, volume 587. John Wiley & Sons.

Kamiya, H. and Eguchi, S. (2001). A class of robust principal component vectors. *Journal of Multivariate Analysis*, 77:239–269.

Kaski, S. and Lagus, K. (1996). Comparing self-organizing maps. In *Proceedings of ICANN 1996*, pages 809–814.

Kégl, B. (2002). Intrinsic dimension estimation using packing numbers. In *Advances in neural information processing systems*, pages 681–688.

Kelley, J. L. (1955). *General topology*, volume 233. van Nostrand, New York, NY.

Kendall, M. G. (1948). *Rank correlation methods.* Griffin London.

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological cybernetics*, 43:59–69.

Kohonen, T. (1990). The self-organizing map. *Proceedings of the IEEE*, 78:1464–1480.

Kouropteva, O., Okun, O., and Pietikäinen, M. (2002). Selection of the optimal parameter value for the locally linear embedding algorithm. In *FSKD*.

Kramer, M. A. (1991). Nonlinear principal component analysis using autoassociative neural networks. *AIChE Journal*, 37:233–243.

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

Kuncheva, L. I. (2004). *Combining pattern classifiers: methods and algorithms.* John Wiley & Sons.

Lambert, D. (1982). Qualitative robustness of tests. *Journal of the American Statistical Association*, 77:352–357.

Lee, J. A. and Verleysen, M. (2007). *Nonlinear Dimensionality reduction.* Springer, New York, NY.

Lee, J. A. and Verleysen, M. (2009). Quality assessment of dimensionality reduction: Rank-based criteria. *Neurocomputing*, 72:1431–1443.

Lee, J. M. (2000). *Introduction to topological manifolds.* Springer, New York, NY.

Lee, J. M. (2003). *Introduction to smooth manifolds.* Springer, New York, NY.

Leng, C., Cheng, J., Yuan, T., Bai, X., and Lu, H. (2014). Learning binary codes with bagging pca. In *Machine Learning and Knowledge Discovery in Databases*, pages 177–192. Springer.

Levina, E. and Bickel, P. J. (2004). Maximum likelihood estimation of intrinsic dimension. In *Advances in neural information processing systems*, pages 777–7847.

Li, G. and Chen, Z. (1985). Projection-pursuit approach to robust dispersion matrices and principal components: Primary theory and monte carlo. *Journal of the American Statistical Association*, 80:759–766.

Li, L., Huang, W., Gu, I. Y.-H., and Tian, Q. (2004). Statistical modeling of complex backgrounds for foreground object detection. *Image Processing, IEEE Transactions on*, 13(11):1459–1472.

Liang, J., Chenouri, S., and Small, C. G. (2015). *Performance Analysis for Dimensionality Reduction.* Electronic Journal of Statistics (submitted manuscript).

Lin, Z., Ganesh, A., Wright, J., Wu, L., Chen, M., and Ma, Y. (2009). Fast convex optimization algorithms for exact recovery of a corrupted low-rank matrix. *Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, 61.

Lindsay, B. G. (1994). Efficiency versus robustness: The case for minimum hellinger distance and related methods. *The Annals of Statistics*, 22:1081–1114.

Liski, E., Nordhausen, K., Oja, H., and Ruiz-Gazen, A. (2012). Averaging orthogonal projectors. *arXiv preprint arXiv:1210.2575*.

Lopuhaa, H. P. (1989). On the relation between s-estimators and m-estimators of multivariate location and covariance. *The Annals of Statistics*, pages 1662–1683.

Lopuhaä, H. P. (1992). Highly efficient estimators of multivariate location with high breakdown point. *The Annals of Statistics*, 20:398–413.

Lopuhaa, H. P. (1999). Asymptotics of reweighted estimators of multivariate location and scatter. *Annals of Statistics*, pages 1638–1665.

Lu, C., Zhang, T., Zhang, R., and Zhang, C. (2003). Adaptive robust kernel pca algorithm. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference*, volume 6, pages VI–621.

Lucas, A. (1997). Asymptotic robustness of least median of squares for autoregressions with additive outliers. *Communications in Statistics - Theory and Methods*, 26:2363–2380.

Lueks, W., Mokbel, B., Biehl, M., and Hammer, B. (2011). How to evaluate dimensionality reduction. In *Proceedings of the workshop–new challenges in neural computation*, volume 5, pages 29–37.

Ma, S., Goldfarb, D., and Chen, L. (2011). Fixed point and bregman iterative methods for matrix rank minimization. *Mathematical Programming*, 128(1-2):321–353.

Ma, Y. and Genton, M. G. (2000). Highly robust estimation of the autocovariance function. *Journal of Time Series Analysis*, 21:663–684.

Mallow, C. L. (1975). On some topics in robustness. Technical report, Bell Telephone Laboratories, Murray Hill, NJ.

Maronna, R. (2005). Principal components and orthogonal regression based on robust scales. *Technometrics*, 47(3):264–273.

Maronna, R. A. (1976). Robust m-estimators of multivariate location and scatter. *The Annals of Statistics*, 4:51–67.

Maronna, R. A., Martin, R. D., and Yohai, V. J. (2006). *Robust Statistics: Theory and Methods*. Wiley, New York, NY.

Mateos, G. and Giannakis, G. B. (2010). Sparsity control for robust principal component analysis. In *Signals, Systems and Computers (ASILOMAR), 2010 Conference Record of the Forty Fourth Asilomar Conference on*, pages 1925–1929. IEEE.

Mekuz, N. and Tsotsos, J. K. (2006). Parameterless isomap with adaptive neighborhood selection. In *Pattern Recognition*, pages 364–373. Springer Berlin Heidelberg.

Mendes, B. V. M. (2000). Assessing the bias of maximum likelihood estimates of contaminated garch models. *Journal of Statistical Computation and Simulation*, 67:359–376.

Milnor, J. W. and Stasheff, J. D. (1974). *Characteristic classes*. Princeton university press.

Müller, C. H. and Uhlig, S. (2001). Estimation of variance components with high breakdown point and high efficiency. *Biometrika*, 88:353–366.

Nadler, B., Lafon, S., Coifman, R. R., and Kevrekidis, I. G. (2005). Diffusion maps, spectral clustering and eigenfunctions of Fokker-Planck operators. *Neural Information Processing Systems*, 18.

Newcomb, S. (1886). A generalized theory of the combination of observations so as to obtain the best result. *American Journal of Mathematics*, 8:343–366.

Nguyen, M. H. and De la Torre, F. (2008). Robust kernel principle component analysis. In *NIPS*.

Papantoni-Kazakos, P. (1984). Some aspects of qualitative robustness in time series. *Robust and nonlinear time series analysis*, 26:218–230. Lecture Notes in Statistics.

Pavan, M. and Pelillo, M. (2007). Dominant sets and pairwise clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29:167–172.

Pearson, K. (1901). On lines and planes of closest fit to sytems of points in space. *Philosophical Magazine*, 2:559–572.

Peres-Neto, P. R., Jackson, D. A., and Somers, K. M. (2005). How many principal components? stopping rules for determining the number of non-trivial axes revisited. *Computational Statistics & Data Analysis*, 49(4):974–997.

Podosinnikova, A., Setzer, S., and Hein, M. (2014). Robust pca: Optimization of the robust reconstruction error over the stiefel manifold. In *Pattern Recognition*, pages 121–131. Springer.

Pölzlbauer, G. (2004). Survey and comparison of quality measures for self-organizing maps. In *Fifth Workshop on Data Analysis (WDA 2004)*, pages 67–82.

Premachandran, V. and Kakarala, R. (2013). Consensus of k-nns for robust neighborhood selection on graph-based manifolds. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE*, pages 1594–1601.

Prendergast, L. A. et al. (2008). A note on sensitivity of principal component subspaces and the efficient detection of influential observations in high dimensions. *Electronic Journal of Statistics*, 2:454–467.

Rieder, H. (1982). Qualitative robustness of rank tests. *The Annals of Statistics*, 10:205–211.

Rousseeuw, P. J. (1984). Least median of squares regression. *Journal of the American Statistical Association*, 79:871–880.

Rousseeuw, P. J. (1985). Multivariate estimation with high breakdown point. *Mathematical statistics and applications*, 8:283–297.

Rousseeuw, P. J. and Croux, C. (1994). The bias of k-step m-estimators. *Statistics & Probability Letters*, 20:411–420.

Rousseeuw, P. J. and Driessen, K. V. (1999). A fast algorithm for the minimum covariance determinant estimator. *Technometrics*, 41(3):212–223.

Rousseeuw, P. J. and Leroy, A. M. (1987). *Robust regression and outlier detection.* Wiley, New York.

Roweis, S. and Saul, L. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326.

Roychowdhury, S. and Ghosh, J. (2009). Robust laplacian eigenmaps using global information. In *Proc. of AAAI Fall Symp on Manifold Learning and Its Applications*.

Ruppert, D. and Carroll, R. (1971). Trimmed least squares estimation in the linear model. *Journal of the American Statistical Association*, 75:828–838.

Sakata, S. and White, H. (1995). An alternative definition of finite sample breakdown point with applications toregression model estimators. *Journal of the American Statistical Association*, 90:1099–1106.

Saul, L. K. and Roweis, S. T. (2003). Think globally, fit locally: unsupervised learning of low dimensional manifolds. *The Journal of Machine Learning Research*, 4:119–155.

Schölkopf, B., Smola, A. J., and Muller, K. R. (1997). Kernel principal component analysis. In *Artificial Neural Networks-ICANN'97*, pages 583–588. Springer Berlin Heidelberg.

Schölkopf, B., Smola, A. J., and Muller, K. R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Journal of Machine Learning Research*, 10:1299–1319.

Serfling, R. J. (1980). *Approximation Theorems of Mathematical Statistics.* Wiley, New York, NY.

Sha, F. and Saul, L. K. (2005). Analysis and extension of spectral methods for nonlinear dimensionality reduction. In *Proceedings of the 22nd international conference on Machine learning*, pages 784–791. ACM.

Shao, C., Huang, H., and Wan, C. (2007). Selection of the suitable neighborhood size for the isomap algorithm. *Neural Networks, IJCNN 2007. International Joint Conference on IEEE*, pages 300–305.

Shao, C. and Wan, C. (2012). Selection of the neighborhood size for manifold learning based on bayesian information criterion. *Journal of Computational Information Systems*, pages 3043–3050.

Shepard, R. N. (1962). The analysis of proximities: multidimensional scaling with unknown distance function. *Psychometrika*, 27:125–140.

Sibson, R. (1978). Studies in the robustness of multidimensional scaling: Procrustes statistics. *Journal of the Royal Statistical Society, B*, 40:234–238.

Sibson, R. (1979). Studies in the robustness of multidimensional scaling: Perturbational analysis of classical scaling. *Journal of the Royal Statistical Society, B*, 41:217–229.

Sibson, R., Bowyer, A., and Osmond, C. (1981). Studies in the robustness of multidimensional scaling: Euclidean models and simulation studies. *Journal of Statistical Computation and Simulation*, 13:273–296.

Sillitto, G. (1947). The distribution of kendall's τ coefficient of rank correlation in rankings containing tie. *Biometrika*, pages 36–40.

Silva, V. D. and Tenenbaum, J. B. (2002). Global versus local methods in nonlinear dimensionality reduction. In *Advances in neural information processing systems*, pages 705–712.

Spence, I. and Ogilvie, J. C. (1989). Robust multidimensional scaling. *Phychometrika*, 54:511–517.

Stahel, W. A. (1981). *Robuste Schätzungen: Infinitesimale Optimalität und Schätzungen von Kovarianzmatrizen.* PhD thesis, ETH Zürich.

Stromberg, A. J. and Ruppert, D. (1992). Breakdown in nonlinear regression. *Journal of the American Statistical Association*, 87:991–997.

Tanaka, Y. (1988). Sensitivity analysis in principal component analysis: influence on the subspace spanned by principal components. *Communications in Statistics-Theory and Methods*, 17(9):3157–3175.

Tanaka, Y. and Castaño-Tostado, E. (1990). Quadratic perturbation expansions of certain functions of eigenvalues and eigenvectors and their application to sensitivity analysis in multivariate methods. *Communications in Statistics-Theory and Methods*, 19(8):2943–2965.

Tang, G. and Nehorai, A. (2011). Robust principal component analysis based on low-rank and block-sparse matrix decomposition. In *Information Sciences and Systems (CISS), 2011 45th Annual Conference on*, pages 1–5. IEEE.

Tenenbaum, J. B., de Silva, V., and Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2323.

Tu, L. W. (2011). *An introduction to manifolds.* Springer, New York, NY, 2 edition.

Tukey, J. W. (1960). A survey of sampling from contaminated distributions. in eds. I. Olkin et al.

Tukey, J. W. (1962). The future of data analysis. *Annals of Mathematical Statistics*, 33:1–67.

Tukey, J. W. (1970). *Exploratory data analysis.* Mimeograph.

Van de Wiel, M. and Di Bucchianico, A. (2001). Fast computation of the exact null distribution of spearman's ρ and page's l statistic for samples with and without ties. *Journal of statistical planning and inference*, 92(1):133–145.

Venna, J. and Kaski, S. (2001). Neighborhood preservation in nonlinear projection methods: An experimental study. In *Proceedings of ICANN 2001*, pages 485–491, Berlin. Springer.

Verboven, S. and Hubert, M. (2005). Libra: a matlab library for robust analysis. *Chemometrics and intelligent laboratory systems*, 75(2):127–136.

Verveer, P. J. and Duin, R. P. (1995). An evaluation of intrinsic dimensionality estimators. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 17(1):81–86.

Villmann, T., Der, R., Herrmann, M., and Martinetz, T. (1997). Topology preservation in self-organizing feature maps: Exact definition and measurement. *IEEE Transactions on Neural Networks*, 8:256–266.

Víšek, J. Á. (1997). Contamination level and sensitivity of robust tests. In *Handbook of Statistics*, 15, chapter 21, pages 633–644. Elsevier Science, North-Holland. Robust Inference.

Weinberger, K. Q. and Saul, L. K. (2006a). An introduction to nonlinear dimensionality reduction by maximum variance unfolding. In *AAAI*, volume 6, pages 1683–1686.

Weinberger, K. Q. and Saul, L. K. (2006b). Unsupervised learning of image manifolds by semidefinite programming. *International Journal of Computer Vision*, 70:77–90.

Welsh, A. H. and Richardson, A. M. (1997). Approaches to the robust estimation of mixed models. In *Handbook of Statistics*, 15, chapter 13, pages 343–384. Elsevier Science, North-Holland. Robust Inference.

Whitney, H. (1936). Differentiable manifolds. *Annals of Mathematics*, 37:645–680.

Wohlberg, B., Chartrand, R., and Theiler, J. (2012). Local principal component pursuit for nonlinear datasets. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pages 3925–3928. IEEE.

Xu, H., Caramanis, C., and Sanghavi, S. (2010). Robust pca via outlier pursuit. In *Advances in Neural Information Processing Systems*, pages 2496–2504.

Yamanishi, Y. and Tanaka, Y. (2006). Sensitivity analysis in kernel principal component analysis. In *COMPSTAT 2006, Proceedings in Computational Statistics*, pages 787–795, Heidelberg. Physica-Verlag.

Yang, X. and Latecki, L. (2011). Affinity learning on a tensor product graph with applications to shape and image retrival. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE*, pages 2369–2376.

Yohai, V. J. (1987). Hgh breakdown-point and high efficiency robust estimates for regression. *The Annals of Statistics*, 15:642–656.

Zhan, Y. and Yin, J. (2009). Robust local tangent space alignment. In *Neural Information Processing*, pages 293–301.

Zhang, Z., Wang, J., and Zha, H. (2012). Adaptive manifold learning. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34:253–265.

Zhang, Z. and Zha, H. (2005). Principal manifolds and nonlinear dimension reduction via local tangent space alignment. *SIAM Journal on Scientific Computing*, 26:313–338.

Zhou, T. and Tao, D. (2011). Godec: Randomized low-rank & sparse matrix decomposition in noisy case. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 33–40. ACM.

Zhou, Z., Li, X., Wright, J., Candes, E., and Ma, Y. (2010). Stable principal component pursuit. In *Information Theory Proceedings (ISIT), 2010 IEEE International Symposium on*, pages 1518–1522. IEEE.

Appendices

A The procedure for solving the transformation matrix

In this appendix we discuss procedures for computing the transformation matrix $\widehat{\mathbf{A}}$ defined in (2.13). Given the input data $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ and the low-dimensional representation $\{\widehat{\mathbf{y}}_1, \ldots, \widehat{\mathbf{y}}_n\}$. We obtain the affine transformation matrix $\widehat{\mathbf{A}}$ as the solution to equation (2.13). It is solved by the following procedure.

(i) To find a proper subset $\mathbb{I}$:

For all cases i, calculate

$$\begin{aligned} d_{ij} &= (\mathbf{x}_i - \mathbf{x}_j)'(\mathbf{x}_i - \mathbf{x}_j), \\ d_i &= \sum_{j \in N_J^I(i)} d_{ij}. \end{aligned}$$

Sort $d_1, \ldots, d_n$, and choose a permutation π such that

$$d_{\pi(1)} \leq d_{\pi(2)} \leq \cdots \leq d_{\pi(n)}.$$

Let $m < n$ be a tuning parameter. We choose the subset $\mathbb{I} = \{\pi(1), \ldots, \pi(m)\}$. In the numerical experiments in this paper, the tuning parameter is set to be $m = \lfloor 0.75n \rfloor$.

(ii) Let $\mathbb{C}^q$ be the set of all $q \times q$ matrix. Given $\mathbb{I}$, we solve the LS solution:

$$\widehat{\mathbf{A}} = \arg\min_{\mathbf{A} \in \mathbb{C}^q} \left\{ \sum_{i \in \mathbb{I}} \sum_{j \in N_J^I(i)} \left[d_{ij} - (\widehat{\mathbf{y}}_i - \widehat{\mathbf{y}}_j)' \mathbf{A}' \mathbf{A} (\widehat{\mathbf{y}}_i - \widehat{\mathbf{y}}_j) \right]^2 \right\}$$

as follows.

- Let $\mathbf{M} = \mathbf{A}'\mathbf{A}$. Define

$$f(\mathbf{M}) = \sum_{i \in \mathbb{I}} \sum_{j \in N_J^I(i)} \left[d_{ij} - (\mathbf{y}_i - \mathbf{y}_j)' \, \mathbf{M} \, (\mathbf{y}_i - \mathbf{y}_j) \right]^2,$$

so that

$$\frac{\partial f}{\partial \mathbf{M}} = \sum_{i \in \mathbb{I}} \sum_{j \in N_J^I(i)} \left\{ 2 \left[d_{ij} - (\widehat{\mathbf{y}}_i - \widehat{\mathbf{y}}_j)' \, \mathbf{M} \, (\widehat{\mathbf{y}}_i - \widehat{\mathbf{y}}_j) \right] (\widehat{\mathbf{y}}_i - \widehat{\mathbf{y}}_j)(\widehat{\mathbf{y}}_i - \widehat{\mathbf{y}}_j)' \right\}$$

- Let $\mathbf{D}_{ij} = (\widehat{\mathbf{y}}_i - \widehat{\mathbf{y}}_j)(\widehat{\mathbf{y}}_i - \widehat{\mathbf{y}}_j)'$, and set $\partial f / \partial \mathbf{M} = 0$, so that

$$\sum_{i \in \mathbb{I}} \sum_{j \in N_J^I(i)} d_{ij} \mathbf{D}_{ij} = \sum_{i \in \mathbb{I}} \sum_{j \in N_J^I(i)} \mathbf{D}_{ij} \, \mathbf{M} \, \mathbf{D}_{ij}. \tag{4.8}$$

- The matrix $\mathbf{M}$ is symmetric, and equation (4.8) can be solved for $\mathbf{M}$ by a linear system.
- Then a proper solution of $\widehat{\mathbf{A}}$ is $\widehat{\mathbf{A}} = \mathbf{M}^{\frac{1}{2}}$.

The computational complexity for step (i) scales as $O(n^2 J)$, and for solving the linear system in step (ii) scales as $O\left(n \left(\frac{q(q+1)}{2}\right)^3\right)$, where q is the dimensionality of the output data $\widehat{\mathbf{Y}}$.

B Distributions of local rank correlations when $\mathbf{X}$ and $\widehat{\mathbf{Y}}$ are independent

Local Kendall correlation $\tau_{_J}$

In this appendix we derive the distribution of local kendall correlation $\tau_{_J}$ when the output $\widehat{\mathbf{Y}}$ are generated independently from the input $\mathbf{X}$. Notice that in this case, the distributions of $\tau^I_{_J}$ and $\tau^O_{_J}$ are the same. Therefore we only consider $\tau^I_{_J}$ here.

- *Distribution of* ζ:

  We first model the overlap size $\zeta = \left| N^I_J(i) \bigcap N^O_J(i) \right|$ by a hypergeometric distribution with J defectives out of $n-1$ items and J draws, i.e.

  $$\Pr(\zeta = r) = \frac{\binom{J}{r}\binom{n-J-1}{J-r}}{\binom{n-1}{J}}.$$

- *Conditional distribution of* $\tau^I_{_J} | \zeta = r$:

  Conditioning on the random overlap, the distribution of $\tau^I_{_J}$ can be derived using the result from Sillitto [1947]. Suppose we have two samples $\{x_{_1}, \ldots, x_{_J}\}$ and $\{y_{_1}, \ldots, y_{_J}\}$ from independent random variables X and Y, respectively. Denote

  $$S_{_J} = \sum_{j<k} \text{sign}\left\{(x_{_j} - x_{_k})(y_{_j} - y_{_k})\right\} .$$

  To derive the general probability mass function of $S_{_J}$, let

  $$g(s, p_{_1}, p_{_2}, \ldots, p_m) = \Pr(S_{_J} = s | p_{_1}, p_{_2}, \ldots, p_m)$$

  be the distribution function of $S_{_J}$ when a ranking of J members containing $p_{_1}$ distinct values, $p_{_2}$ pairs, $p_{_3}$ triplets, ... , and p_m m-tuplets. The exact pmf of $S_{_J}$ for sample size J from 3 to 7, and for ranking containing only distinct values and p_2 pairs is provided (in a table) in Sillitto [1947]. It also provided a recursive formula

$$\begin{aligned} g(s, p_1, p_2, \ldots, p_m) &= g(s-(m-1), p_1 - 1, p_2, \ldots, p_{m-1} - 1, p_m + 1) \\ &+ g(s-(m-3), p_1 - 1, p_2, \ldots, p_{m-1} - 1, p_m + 1) \\ &\cdots \\ &+ g(s+(m-1), p_1 - 1, p_2, \ldots, p_{m-1} - 1, p_m + 1) \,. \end{aligned}$$

Now, we get back to the definition of τ_J. Let

$$T(i) = \sum_{j<k \in N_J^I(i)} \operatorname{sign}\left\{[\widehat{R}_{ij} - \widehat{R}_{ik}] \cdot (s_{ij} - s_{ik})\right\} \,.$$

Conditioning on $\zeta = r$, $\{s_{i1}, \ldots, s_{iJ}\}$ contains no ties, and there is one $(J-r)$-tuplet in $\left\{\widehat{R}_{i1}, \ldots, \widehat{R}_{iJ}\right\}$. Thus, the conditional pmf of $T(i)|\zeta = r$ is a special case of the above result, and it can be calculated from the recursive formula.

$$\Pr(T(i) = t | \zeta = r) = g(t, p_1, p_2, \ldots, p_{J-r}) \,,$$

where $p_1 = r$, $p_{J-r} = 1$, and $p_m = 0$ for $m = 2, \ldots, J-r-1$. Recall that

$$\tau_J^I(i) = \frac{T(i)}{\frac{1}{2}J(J-1)}.$$

Since $\frac{1}{2}J(J-1)$ is a constant for a fixed J, the conditional distribution of $\tau_J^I|\zeta = r$ can be easily obtained from $\Pr(T(i) = t|\zeta = r)$. We denote the conditional pmf by

$$f_r(z) = \Pr(\tau_J^I = z | \zeta = r) \,.$$

- *Distribution of τ_J^I:*

Denote $f(z) = \Pr(\tau_{_J}^I = z)$.

$$\begin{aligned} f(z) &= \sum_{r=0}^{J} \Pr(\tau_{_J}^I = z|\zeta = r) \cdot \Pr(\zeta = r) \\ &= \sum_{r=0}^{J} \left\{ f_r(z) \cdot \frac{\binom{J}{r}\binom{n-J-1}{J-r}}{\binom{n-1}{J}} \right\}. \end{aligned}$$

- *Expectation and variance of* $\tau_{_J}^I$:

  Assuming independence between $\mathbf{X}$ and $\widehat{\mathbf{Y}}$, it can be easily seen that $\mathrm{E}(T(i)|\zeta = r) = 0$. This implies that $\mathrm{E}(\tau_{_J}^I) = 0$.

  Now applying the result in Sillitto [1947], we can obtain the conditional variance

$$\mathrm{Var}(T(i)|\zeta = r) = \begin{cases} r(6J^2 - 6Jr + 6J + 2r^2 - 3r - 5)/18, & 0 \le r \le J-2 \\ J(J-1)(2J+5)/18, & r = J-1, J \end{cases}$$

$$\mathrm{Var}(\tau_{_J}^I|\zeta = r) = \begin{cases} \frac{2r(6J^2-6Jr+6J+2r^2-3r-5)}{9J^2(J-1)^2}, & 0 \le r \le J-2 \\ \frac{2(2J+5)}{9J(J-1)}, & r = J-1, J. \end{cases}$$

  Thus

$$\begin{aligned} \mathrm{Var}(\tau_{_J}^I) &= \mathrm{E}\left\{\mathrm{Var}(\tau_{_J}^I|\zeta = r)\right\} + \mathrm{Var}\left\{\mathrm{E}(\tau_{_J}^I|\zeta = r)\right\} \\ &= \mathrm{E}\left\{\mathrm{Var}(\tau_{_J}^I|\zeta = r)\right\} \\ &= \sum_{r=0}^{J} \left\{ \mathrm{Var}(\tau_{_J}^I|\zeta = r) \cdot \frac{\binom{J}{r}\binom{n-J-1}{J-r}}{\binom{n-1}{J}} \right\}. \end{aligned}$$

Local Spearman correlation $\rho_{_J}$

Similar to Kendall's $\tau_{_J}$, we can derive the distribution of the local Spearman correlation $\rho_{_J}$. Again we will consider only the case $\rho_{_J}^I$ because when $\widehat{\mathbf{Y}}$ is generated independently from $\mathbf{X}$, the distributions of $\rho_{_J}^I$ and $\rho_{_J}^O$ are the same.

- *Conditional distribution of* $\rho^I_J|\zeta$:

  Rewrite ρ^I_J as

$$\begin{aligned}\rho^I_J(i,\mathbf{X},\mathbf{Y}) &= 1 - \frac{\sum\limits_{j\in N^I_J(i)} \left(s_{ij} - \widehat{R}_{ij}\right)^2 + U}{\frac{1}{6}(J^3 - J)} \\ &= 1 - \frac{4J+2}{J-1} + \frac{\sum\limits_{j\in N^I_J(i)} \left(s_{ij} \cdot \widehat{R}_{ij}\right)}{\frac{1}{12}(J^3 - J)} \,.\end{aligned}$$

  Let

$$T_\rho(i) = \sum_{j\in N^I_J(i)} \left(s_{ij} \cdot \widehat{R}_{ij}\right) \,.$$

  The conditional distribution of $T_\rho(i)|\zeta$ can be calculated via its probability generating function using the result in Van de Wiel and Di Bucchianico [2001]. However, it is difficult to write out the explicit form of the conditional distribution of ρ^I_J because there is no recursive formula available.

- *Expectation and variance of* ρ^I_J:

  Conditioning on $\zeta = r$, s_{ij} takes value from 1 to J, and $\widehat{R}_{ij}$ takes value in

$$\left\{1, \ldots, r, \underbrace{\frac{r+J+1}{2}, \ldots, \frac{r+J+1}{2}}_{J-r}\right\}.$$

  Therefore we can write

$$\begin{aligned}T_\rho(i) &= \sum_{j\in N^I_J(i)} \left(s_{ij} \cdot \widehat{R}_{ij}\right) \\ &= \sum_{k=1}^{J} \left(k \cdot \sigma(k)\right) \,.\end{aligned}$$

where

$$\sigma(k) \in \left\{ 1, \dots, r, \underbrace{\frac{r+J+1}{2}, \dots, \frac{r+J+1}{2}}_{J-r} \right\}.$$

Assuming independence between $\mathbf{X}$ and $\widehat{\mathbf{Y}}$, for any k, we have

$$\begin{aligned} &\Pr(\sigma(k) = j|\zeta = r) = 1/J,\ j = 1, \dots, r \\ &\Pr(\sigma(k) = (r+J+1)/2|\zeta = r) = (J-r)/J. \end{aligned}$$

Thus, we can show that

$$\begin{aligned} \mathrm{E}\left(T_\rho(i)|\zeta = r\right) &= \frac{1}{J}\sum_{k=1}^{J}\left\{k \cdot \left((J-r)\cdot\frac{r+J+1}{2} + \sum_{j=1}^{r} j\right)\right\} \\ &= \frac{1}{4}J(J+1)^2. \end{aligned}$$

Similarly we have

$$\mathrm{E}\left(T_\rho^2(i)|\zeta = r\right) = \frac{1}{144}J(J+1)(9J^4 + 27J^3 + 3rJ^2 + 9J - 3r^2J - r + r^3).$$

Then the expected value of $\rho_{_J}(i)$ can be obtained by

$$\begin{aligned} \mathrm{E}\left(\rho_{_J}(i)\right) &= 1 - \frac{4J+2}{J-1} - \frac{\mathrm{E}\left(T_\rho(i)\right)}{\frac{1}{12}(J^3-1)} \\ &= 1 - \frac{4J+2}{J-1} - \frac{\frac{1}{4}J(J+1)^2}{\frac{1}{12}(J^3-J)} \\ &= 0, \end{aligned}$$

and the variance of $\rho_{_J}(i)$ is obtained by

$$\begin{aligned}
\mathrm{Var}\left(T_\rho(i)|\zeta=r\right) &= \mathrm{E}\left(T_\rho^2(i)|\zeta=r\right)-\left\{\mathrm{E}\left(T_\rho(i)|\zeta=r\right)\right\}^2 \\
\mathrm{Var}\left(\rho_{_J}(i)|\zeta=r\right) &= \frac{\mathrm{Var}\left(T_\rho(i)|\zeta=r\right)}{\left(\frac{1}{12}(J^3-J)\right)^2} \\
&= \frac{r(3J^3-3rJ+r^2-1)}{J(J+1)(J-1)^2}
\end{aligned}$$

$$\mathrm{Var}\left(\rho_{_J}(i)\right)=\sum_{r=0}^{J}\left\{\mathrm{Var}\left(\rho_{_J}(i)|\zeta=r\right)\cdot\frac{\binom{J}{r}\binom{n-J-1}{J-r}}{\binom{n-1}{J}}\right\}$$