Lecture Notes
in Computational Science
and Engineering

58

Editors

Alexander N. Gorban Balázs Kégl
Donald C. Wunsch Andrei Zinovyev
(Eds.)

Principal Manifolds for Data Visualization and Dimension Reduction

With 82 Figures and 22 Tables

Springer

Editors

Alexander N. Gorban
Department of Mathematics
University of Leicester
University Road, LE1 7RH
Leicester, United Kingdom
email: ag153@leicester.ac.uk

Donald C. Wunsch
Department of Electrical
and Computer Engineering
University of Missouri - Rolla
1870 Miner Circle
Rolla, Missouri 65409, USA
email: dwunsch@ece.umr.edu

Balázs Kégl
University of Paris-Sud - CNRS
Linear Accelerator Laboratory
Bâtiment 2000
91898 Orsay, France
email: kegl@lal.in2p3.fr

Andrei Zinovyev
Institut Curie
Service Bioinformatique
rue d'Ulm 26, 75248
Paris, France
email: andrei.zinovyev@curie.fr

Library of Congress Control Number: 2007932175

Mathematics Subject Classification: 62H25, 62H30, 62P10, 62P35, 68Q85, 68T10, 68T05, 68U05, 92D10, 57-06

ISBN 978-3-540-73749-0 Springer Berlin Heidelberg New York

Typesetting by the editors and SPi using a Springer LaTeX macro package
Cover design: WMX Design GmbH, Heidelberg

Printed on acid-free paper SPIN: 10825826 46/SPi 5 4 3 2 1 0

Preface

In 1901, Karl Pearson [1] explained to the scientific community that the problem of data approximation is (i) important and (ii) nice, and (iii) differs from the regression problem. He demonstrated how to approximate data sets with straight lines and planes. That is, he invented Principal Component Analysis (PCA). Why and when do we need to solve the data approximation problem instead of regression? Let us look at Pearson's explanation:

> "(1) In many physical, statistical, and biological investigations it is desirable to represent a system of points in plane, three, or higher dimensioned space by the "best-fitting" straight line or plane. Analytically this consists in taking
>
> $$y = a_0 + a_1 x, \quad \text{or} \quad z = a_0 + a_1 x + b_1 y,$$
> $$\text{or} \quad z = a_0 + a_1 x_1 + a_2 x_2 + a_3 x_3 + \ldots + a_n x_n,$$
>
> where y, x, z, x_1, x_2, $\ldots x_n$ are variables, and determining the "best" values for constants a_0, a_1, b_1, a_0, a_1, $a_2, \ldots a_n$ in relation to the observed corresponding values of the variables. In nearly all the cases dealt with in the text-books of least squares, the variables on the right of our equations are treated as the independent, those on the left as the dependent variables. The result of this treatment is that we get one straight line or plane if we treat some one variable as independent, and a quite different one if we treat another variable as the independent variable. There is no paradox about this; it is, in fact, an easily understood and most important feature of the theory of a system of correlated variables. The most probable value of y for a given value of x, say, is not given by the same relation as the most probable value of x for the given value of y. Or, to take a concrete example, the most probable stature of a man with a given length of leg l being s, the most probable length of leg for a man with stature s will not be l. The "best-fitting" lines and planes ... depend upon

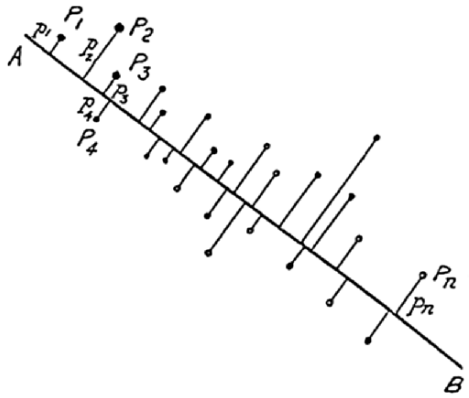

Fig. 1. Data approximation by a straight line. The famous illustration from Pearson's paper [1]

a determination of the means, standard-deviations, and correlation-coefficients of the system. In such cases the values of the independent variables are supposed to be accurately known, and the probable value of the dependent variable is ascertained.

(2) In many cases of physics and biology, however, the "independent" variable is subject to just as much deviation or error as the "dependent" variable. We do not, for example, know x accurately and then proceed to find y, but both x and y are found by experiment or observation. We observe x and y and seek for a unique functional relation between them. Men of given stature may have a variety of leg-length; but a point at a given time will have one position only, although our observation of *both* time and position may be in error, and vary from experiment to experiment. In the case we are about to deal with, we suppose the observed variables – all subject to error – to be plotted in plane, three-dimensioned or higher space, and we endeavour to tale a line (or plane) which will be the "best fit" to such system of points.

Of course the term "best fit" is really arbitrary; but a good fit will clearly be obtained if we make the sum of the squares of the perpendiculars from the system of points upon the line or plane a minimum.

For example:—Let P_1, P_2, $\ldots P_n$ be the system of points with co-ordinates x_1, y_1; x_2, y_2; $\ldots$ x_n, y_n, and perpendicular distances p_1, $p_2, \ldots p_n$ from a line **AB**. Then we shall make[1]

$$U = S(p^2) = \text{a minimum."}$$

[1] $S(p^2)$ stands for $\sum_i p_i^2$; for details see Fig. 1

This explanation sounds very modern: in "many cases of physics and biology" there is significant noise in the "independent variables", and it appears better to approximate data points than the regression functions that transform one set of data coordinates (the "independent variables") into another. "Of course the term "best fit" is really arbitrary", but the least squares approach remains the method of choice, if there exist no strong arguments for another choice of metrics. This method was applied to many problems, has been transformed and rediscovered several times, and is now known under several names: mostly as PCA or as *proper orthogonal decomposition*. But the main idea remains the same: we approximate the data set by a point (this is the mean point), then by a line (first principal component), then by a plane, etc.

What was invented in the data approximation during the century? First of all, the approximation by linear manifolds (lines, planes, ...) was supplemented by a rich choice of the approximate objects. The important discovery is the approximation of a data set by a smaller finite set of "centroids". In the least squares approach to the best fit this gives the famous K-means algorithm [2]. Usually, this method is discussed as a clustering algorithm, but its application field is much wider. It is useful for adaptive coding and data binning, and is a model reduction method, as well as the PCA: the PCA allows us to substitute a high-dimensional vector by its projection on a best fitted low-dimensional linear manifold, the K-means approach gives an approximation of a big data set by K best fitted centroids.

Between the "most rigid" linear manifolds and "most soft" unstructured finite sets there is the whole universe of approximants. If we change the PCA linear manifolds to algebraic curves and manifolds, then a branch of the *algebraic statistics* appears. This field is still relatively new (less than ten years old) [3]. Algebraic curves and manifolds are much more flexible than linear ones, but remain rigid in the following sense: it is impossible to change the algebraic manifold locally, only near a point. Differential manifolds give more freedom, but require specific efforts for regularization.

A step from absolute flexibility of finite sets gives the Self-Organizing Map (SOM) approach [4]. SOM can be formalized either as a manifold learner which represents the manifold as a discretized grid, or a K-means-like clustering algorithm which adds a topology to the cluster centroids. Although SOM has been slowly replaced by theoretically better founded and better behaving algorithms, its simplicity and computational efficiency makes it one of the most popular data analysis techniques even today. An important improvement of SOM came with the introduction of the Generative Topographic Mapping (GTM) [6], establishing a probabilistic framework and a well-defined objective function. The generative probabilistic model has also become an analytical tool to formalize the faithfulness-conciseness trade-off.

Another big shift of the century is the appearance of the whole framework of machine learning which significantly extends Pearson's initial "geometrical" approach. It is a common practice in general discussions on machine learning

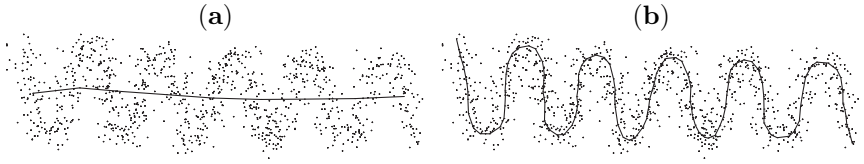

Fig. 2. An ill-defined unsupervised learning problem. Which curve describes the data better, (**a**) a short curve that is "far" from the data, or (**b**) a long curve that follows the data closely?

to use the dichotomy of supervised and unsupervised learning to categorize learning methods. Supervised learning algorithms assume that a training set of *(input, output)* observations is given (e.g., digitized images of characters and their class labels). The goal is then to learn a function that predicts the output for previously unseen input patterns. This is a very far generalization of Pearson's linear regression onto various types of inputs, outputs, and functions.

In unsupervised learning, we only have a set of (input) observations without a desired target, and the goal is to find an efficient representation of the data (for example by reducing the number of attributes or grouping the data into a small number of clusters), or to characterize the data-generating process.

From a conceptual point of view, unsupervised learning is substantially more difficult than supervised learning. Whereas in supervised learning the cost of mis-predicting the output provides a well-defined criteria to optimize, in unsupervised learning we often face a trade-off of representing the data as faithfully as possible while being as concise as possible (Fig. 2). In a certain sense, an unsupervised learner can be considered as a supervised learner where the target is the input itself. In other words, the task is to find a function as close to the identity function as possible. Of course, without restricting the set of admissible predictors this is a trivial problem. These restrictions originate from the other objective of unsupervised learning of finding a mapping which is simple in a certain sense. The trade-off between these two competing objectives depends on the particular problem.

Manifold learning is a sub-domain of unsupervised learning where the goal is to project the input data into a new space which is simpler in a certain sense than the input space, or in which the data distribution is more regular than originally.

Two distinct groups of methods exist for this purpose that differ in their way of representing the manifold. Thus, Non-linear PCA (NLPCA) extends PCA by replacing the linear encoder and decoder by non-linear functions (for example, feed-forward neural networks [7]), and optimizing them in an auto-encoding setup. The embedded manifold appears only implicitly as the decoded image of the input space, and the geometric notion of projection does not apply.

Principal curves and manifolds [8], on the other hand, extend the geometric interpretation of PCA by explicitly constructing an embedded manifold, and by encoding using standard geometric projection onto the manifold. How to define the "simplicity" of the manifold is problem-dependent, however, it is commonly measured by the intrinsic dimensionality and/or the smoothness of the manifold.

Clustering, another important sub-domain of unsupervised learning, can also be formalized in this framework: the clustering "manifold" is a finite partitioning of the input space, in the simplest case represented as a finite set of singular centroid points. Obviously, in this case simplicity can be measured neither by smoothness nor dimensionality, nevertheless, manifold learning and clustering methodologies are intimately connected both in their theoretical underpinning and on a technical-algorithmic level.

Most of the modern manifold learners find their theoretical and algorithmic roots in one of three basic and well-known data analysis techniques: PCA, K-means, and Multidimensional Scaling (MDS) [5] also known as Torgerson or Torgerson-Gower scaling. Thus, the basic loop of K-means that alternates between a projection and an optimization step became the algorithmic skeleton of many non-linear manifold learning algorithms. The SOM algorithm is arguably the torch holder of this batch of nonlinear manifold learners.

The objective of original MDS is somewhat different: find a linear projection that preserves pairwise distances as well as possible. The method does not explicitly construct an embedded manifold, but it has the important role of being the algorithmic forefather of "one-shot" (non-iterative) manifold learners. The most recent representatives of this approach are Local Linear Embedding (LLE) [9] and ISOMAP [10]. Both methods find their origins in MDS in the sense that their goal is to preserve pairwise relationships between data points. LLE conserves local linear patterns whereas ISOMAP applies MDS using the geodesic (manifold) distance approximated by the shortest path on the neighborhood graph (the graph constructed by connecting nearby points). Since the birth of these two methods, several neighborhood-graph-based techniques have emerged, stimulating the development of a common theory around Laplacian eigenmaps and spectral clustering and embedding.

Despite the significant progress made in the last decade, the manifold learning problem is far from being solved. The main drawback of iterative methods is that they are sensitive to initialization, and they can be stuck easily in suboptimal local minima, especially if the manifold is "loopy" or has a complicated topology. Neighborhood-graph-based "one-shot" techniques behave much better in this respect, their disadvantages are computational inefficiency (the complexity of the construction of the neighborhood graph by itself is quadratic in the number of data points) and increased sensitivity to noise around the manifold. One of today's challenges in manifold learning is to find techniques that combine the advantages of these often incompatible approaches. Another exciting area is non-local manifold learning [11], which abandons two of the implicit premises of manifold learning: that manifolds are

smooth (locally linear) and that we have enough observations in every neighborhood to locally estimate the manifold. A third, very recent but promising, new domain is building deep networks (multiply nested functions) using an unsupervised paradigm (building all the layers except for the last using, for example, an autoassociative objective function [12]). These new areas share the ambitious goal of embedding manifold learning into artificial intelligence in a broad sense.

This book is a collection of reviews and original papers presented partially at the workshop "Principal manifolds for data cartography and dimension reduction" (Leicester, August 24-26, 2006). The problems of Large Data Sets analysis and visualisation, model reduction and the struggle with complexity of data sets are important for many areas of human activity. There exist many scientific and engineering communities that attack these problems from their own sides, and now special efforts are needed to organize communication between these groups, to support exchange of ideas and technology transfer among them. Heuristic algorithms and seminal ideas come from all application fields and from mathematics also, and mathematics has a special responsibility to find a solid basis for heuristics, to transform ideas into exact knowledge, and to transfer the resulting ideal technology to all the participants of the struggle with complexity. The workshop was focused on modern theory and methodology of geometric data analysis and model reduction. Mathematicians, engineers, software developers and advanced users from different areas of applications attended this workshop.

The first chapter of the book presents a general review of existing NLPCA algorithms (U. Kruger, J. Zhang, and L. Xie). Next, M. Scholz, M. Fraunholz, and J. Selbig focus attention on autoassociative neural network approach for NLPCA with applications to metabolite data analysis and gene expression analysis. H. Yin provides an overview on the SOM in the context of manifold learning. Its variant, the visualisation induced SOM (ViSOM) proposed for preserving local metric on the map, is introduced and reviewed for data visualisation. The relationships among the SOM, ViSOM, multidimensional scaling, and principal curves are analysed and discussed. A. Gorban and A. Zinovyev developed a general geometric framework for constructing "principal objects" of various dimensions and topologies with the simple quadratic form of the smoothness penalty. The approach was proposed in the middle of 1990s. It is based on mechanical analogy between principal manifolds and elastic membranes and plates.

M. Peña, W. Barbakh, and C. Fyfe present a family of topology preserving mappings similar to SOM and GTM. These techniques can be considered as a non-linear projection from input or data space to the output or latent space. B. Mirkin develops the iterative extraction approach to clustering and describes additive models for clustering entity-to-feature and similarity. This approach emerged within the PCA framework by extending the bilinear Singular Value Decomposition model to that of clustering.

In their contribution, J. Einbeck, L. Evers, and C. Bailer-Jones give a short review of localized versions of PCA, focusing on local principal curves and local partitioning algorithms. These methods can work with branched and disconnected principal components. S. Girard and S. Iovleff introduce auto-associative models, a new tool for building NLPCA methods, and compare it to other modern methods. A. Gorban, N. Sumner, and A. Zinovyev propose new type of low-dimensional "principal object": *principal cubic complex*, the product of one-dimensional branching principal components. This complex is a generalization of linear and non-linear principal manifolds and includes them as a particular case. To construct such an object, they combine the method of *topological grammars* with the minimization of elastic energy defined for its embedding into multidimensional data space.

B. Nadler, S. Lafon, R. Coifman, and I. G. Kevrekidis provide a diffusion based probabilistic analysis of embedding and clustering algorithms that use the normalized graph Laplacian. They define a random walk on the graph of points and a diffusion distance between any two points. The characteristic relaxation times and processes of the random walk on the graph govern the properties of spectral clustering and spectral embedding algorithms. Specifically, for spectral clustering to succeed, a necessary condition is that the mean exit times from each cluster need to be significantly larger than the largest (slowest) of all relaxation times inside all of the individual clusters. Diffusion metrics is studied also by S. Damelin in the context of the optimal discretization problem. He shows that a general notion of extremal energy defines a diffusion metric on X which is equivalent to a discrepancy on X. The diffusion metric is used to learn X via normalized graph Laplacian dimension reduction and the discepancy is used to discretize X.

Modern biological applications inspire development of new approaches to data approximation. In many chapters biological applications play central role. For the comparison of various algorithms, several test datasets were selected and presented to the workshop participants. These datasets contain results of a high-throughput experimental technology application in molecular biology (microarray data). Principal component analysis and principal manifolds are useful methods for analysis of this kind of data, where the "curse of dimensionality" is an important issue. Because of it some variant of dimension reduction is absolutely required, for example, for regularization of classification problems that simply can not be solved otherwise. An interesting and underexplored question is: can non-linear principal manifolds serve better for this purpose as compared to the linear PCA or feature preselection?

M. Journée, A. E. Teschendorff, P.-A. Absil, S. Tavaré, and R. Sepulchre present an overview of the most popular algorithms to perform ICA. These algorithms are then applied on a microarray breast-cancer data set. D. Elizondo, B. N. Passow, R. Birkenhead, and A. Huemer present a comparison study of the performance of the linear principal component analysis and the non linear local tangent space alignment principal manifold methods to the problem of dimensionality reduction of microarray data.

The volume ends with a tutorial "PCA and K-Means decipher genome". This exercise on principal component analysis and K-Means clustering can be used for courses of statistical methods and bioinformatics. By means of PCA students "discover" that the information in the genome is encoded by non-overlapping triplets. Next, they learn to find gene positions. In Appendix the MatLab program listings are presented.

The methods of data approximation, data visualization and model reduction developed during last century form an important part of the modern intellectual technology of data analysis and modeling. In this book we present some slices of this interdisciplinary technology and aim at eliminating some of the traditional language barriers that, unnecessarily sometimes, impede scientific cooperation and interaction of researchers across disciplines.

References

1. Pearson, K.: On lines and planes of closest fit to systems of points in space. Philosophical Magazine, Ser. VI 2, 559–572 (1901)
2. MacQueen, J. B.: Some methods for classification and analysis of multivariate observations. In: Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability, vol. 1. University of California Press, Berkeley, 281–297 (1967)
3. Pachter, L., Sturmfels, B. (eds): Algebraic Statistics for Computational Biology, Cambridge University Press, Cambridge, United Kingdom (2005)
4. Kohonen, T.: The Self-Organizing Map. Springer, Berlin Heidelberg New York (1997)
5. Torgerson, W. S.: Theory and Methods of Scaling. Wiley, New York (1958)
6. Bishop, C. M., Svensén, M., and Williams, C. K. I.: The generative topographic mapping. Neural Computation, **10**, 215–235 (1998)
7. Kramer, M.A.: Nonlinear principal component analysis using autoassociative neural networks. AIChE Journal, **37**, 233–243 (1991)
8. Hastie, T. and Stuetzle, W.: Principal curves. Journal of the American Statistical Association, **84**, 502–516 (1989)
9. Roweis, S. and Saul L. K.: Nonlinear dimensionality reduction by locally linear embedding. Science, **290**, 2323–2326 (2000)
10. Tenenbaum, J. B., de Silva, V., and Langford J. C.: A global geometric framework for nonlinear dimensionality reduction. Science, **290**, 2319–2323 (2000)
11. Bengio, Y., Monperrus, M., and Larochelle, H.: Nonlocal estimation of manifold structure. Neural Computation, **18**, 2509–2528 (2006)
12. Hinton, G. E. and Salakhutdinov, R.: Reducing the dimensionality of data with neural networks. Science, **313**, 504–507 (2006)

Leicester, UK — *Alexander N. Gorban*
Orsay, France — *Balázs Kégl*
Rolla, MO, USA — *Donald C. Wunsch*
Paris, France — *Andrei Y. Zinovyev*

May, 2007

Contents

List of Contributors

Pierre-Antoine Absil
Department of Mathematical Engineering
Université catholique de Louvain
Batiment Euler - Parking 13
Av. Georges Lemaitre 4
1348 Louvain-la-Neuve
Belgium

Coryn Bailer-Jones
Max-Planck-Institut für Astronomie
Königstuhl 17
69117 Heidelberg
Germany
calj@mpia-hd.mpg.de

Wesam Barbakh
Applied Computational Intelligence Research Unit
The University of Paisley
Paisley PA1 2BE
Scotland
United Kingdom
wesam.barbakh@paisley.ac.uk

Ralph Birkenhead
Centre for Computational Intelligence
School of Computing
Faculty of Computing Sciences and Engineering
De Montfort University
The Gateway
Leicester LE1 9BH
United Kingdom
rab@dmu.ac.uk

Ronald Coifman
Department of Mathematics
Yale University
New Haven, CT 06520-8283
USA
coifman@math.yale.edu

Steven B. Damelin
Department of Mathematical Sciences
Georgia Southern University
PO Box 8093
Statesboro, GA 30460
USA
damelin@georgiasouthern.edu

Jochen Einbeck
Department of Mathematical Sciences
Durham University
Science Laboratories
South Road
Durham DH1 3LE
United Kingdom
jochen.einbeck@durham.ac.uk

David A. Elizondo
Centre for Computational Intelligence
School of Computing
Faculty of Computing Sciences and Engineering
De Montfort University
The Gateway
Leicester LE1 9BH
United Kingdom
elizondo@dmu.ac.uk

Ludger Evers
Department of Mathematics
University of Bristol
University Walk
Bristol BS8 1TW
United Kingdom
l.evers@bris.ac.uk

Martin Fraunholz
Competence Centre for Functional Genomics
Institute for Microbiology
Ernst-Moritz-Arndt-University Greifswald
F.-L.-Jahn-Str. 15
17487 Greifswald
Germany
Martin.Fraunholz@uni-greifswald.de

Colin Fyfe
Applied Computational Intelligence Research Unit
The University of Paisley
Paisley PA1 2BE
Scotland
United Kingdom
colin.fyfe@paisley.ac.uk

Stéphane Girard
INRIA Rhône-Alpes
Projet Mistis
Inovallée 655
av. de l'Europe
Montbonnot 38334 Saint-Ismier cedex
France
Stephane.Girard@inrialpes.fr

Alexander N. Gorban
Department of Mathematics
University of Leicester
University Road
Leicester LE1 7RH
United Kingdom
and
Institute of Computational Modeling
Russian Academy of Sciences
Siberian Branch
Krasnoyarsk 660036
Russia
ag153@le.ac.uk

Andreas Huemer
Centre for Computational Intelligence
School of Computing
Faculty of Computing Sciences and Engineering
De Montfort University
The Gateway
Leicester LE1 9BH
United Kingdom
ahuemer@dmu.ac.uk

Serge Iovleff
Laboratoire Paul Painlevé
59655 Villeneuve d'Ascq Cedex
France
serge.iovleff@univ-lille1.fr

Michel Journée
Department of Electrical Engineering and Computer Science
University of Liège
B-4000 Liège Sart-Tilman
Belgium

Ioannis G. Kevrekidis
Department of Chemical Engineering
and Program in Applied
and Computational Mathematics
Princeton University
Princeton, NJ 08544
USA
yannis@princeton.edu

Uwe Kruger
School of Electronics
Electrical Engineering
and Computer Science
Queen's University Belfast
Belfast BT9 5AH
United Kingdom
uwe.kruger@ee.qub.ac.uk

Stephane Lafon
Department of Mathematics
Yale University
New Haven, CT 06520-8283
USA
stephane.lafon@yale.edu
and Google Inc.

Boris Mirkin
School of Computer Science
and Information Systems
Birkbeck College
University of London
Malet Street
London WC1E 7HX
United Kingdom
mirkin@dcs.bbk.ac.uk

Boaz Nadler
Department of Computer Science
and Applied Mathematics
Weizmann Institute of Science
Rehovot 76100
Israel
boaz.nadler@weizmann.ac.il

Benjamin N. Passow
Centre for Computational
Intelligence
School of Computing
Faculty of Computing Sciences
and Engineering
De Montfort University
The Gateway
Leicester LE1 9BH
United Kingdom
passow@dmu.ac.uk

Marian Peña
Applied Computational
Intelligence Research Unit
The University of Paisley
Paisley PA1 2BE
Scotland
United Kingdom
marian.pena@paisley.ac.uk

Matthias Scholz
Competence Centre
for Functional Genomics
Institute for Microbiology
Ernst-Moritz-Arndt-University
Greifswald
F.-L.-Jahn-Str. 15
17487 Greifswald
Germany
Matthias.Scholz
@uni-greifswald.de

Joachim Selbig
Institute for Biochemistry
and Biology
University of Potsdam
c/o Max Planck Institute
for Molecular Plant Physiology
Am Mühlenberg 1
14424 Potsdam
Germany
Selbig@mpimp-golm.mpg.de

Rodolphe Sepulchre
Department of Electrical Engineering
and Computer Science
University of Liège
B-4000 Liège Sart-Tilman
Belgium
r.sepulchre@ulg.ac.be

Neil R. Sumner
Department of Mathematics
University of Leicester
University Road
Leicester LE1 7RH
United Kingdom
nrs7@le.ac.uk

Simon Tavaré
Breast Cancer
Functional Genomics Program
Cancer Research UK
Cambridge Research Institute
Department of Oncology
University of Cambridge
Robinson Way
Cambridge CB2 0RE
United Kingdom

Andrew E. Teschendorff
Breast Cancer
Functional Genomics Program
Cancer Research UK
Cambridge Research Institute
Department of Oncology
University of Cambridge
Robinson Way
Cambridge CB2 0RE
United Kingdom

Lei Xie
National Key Laboratory
of Industrial Control Technology
Zhejiang University
Hangzhou 310027
P.R. China
leix@iipc.zju.edu.cn

Hujun Yin
School of Electrical
and Electronic Engineering
The University of Manchester
Manchester M60 1QD
United Kingdom
hujun.yin@manchester.ac.uk

Junping Zhang
Department of Computer Science
and Engineering
Fudan University
Shanghai 200433
P.R. China
jpzhang@fudan.edu.cn

Andrei Y. Zinovyev
Institut Curie
26 rue d'Ulm
Paris 75248
France
and
Institute of Computational Modeling
Russian Academy of Sciences
Siberian Branch
Krasnoyarsk 660036
Russia
andrei.zinovyev@curie.fr

1

Developments and Applications of Nonlinear Principal Component Analysis – a Review

Uwe Kruger[1], Junping Zhang[2], and Lei Xie[3]

[1] School of Electronics, Electrical Engineering and Computer Science, Queen's University Belfast, BT9 5AH, UK, uwe.kruger@ee.qub.ac.uk
[2] Department of Computer Science and Engineering, Fudan University, Shanghai 200433, P.R. China, jpzhang@fudan.edu.cn
[3] National Key Laboratory of Industrial Control Technology, Zhejiang University, Hangzhou 310027, P.R. China, leix@iipc.zju.edu.cn

Summary. Although linear principal component analysis (PCA) originates from the work of Sylvester [67] and Pearson [51], the development of nonlinear counterparts has only received attention from the 1980s. Work on nonlinear PCA, or NLPCA, can be divided into the utilization of autoassociative neural networks, principal curves and manifolds, kernel approaches or the combination of these approaches. This article reviews existing algorithmic work, shows how a given data set can be examined to determine whether a conceptually more demanding NLPCA model is required and lists developments of NLPCA algorithms. Finally, the paper outlines problem areas and challenges that require future work to mature the NLPCA research field.

1.1 Introduction

PCA is a data analysis technique that relies on a simple transformation of recorded observation, stored in a vector $\mathbf{z} \in \mathbb{R}^N$, to produce statistically independent score variables, stored in $\mathbf{t} \in \mathbb{R}^n$, $n \leq N$:

$$\mathbf{t} = \mathbf{P}^T \mathbf{z} \, . \tag{1.1}$$

Here, $\mathbf{P}$ is a transformation matrix, constructed from *orthonormal* column vectors. Since the first applications of PCA [21], this technique has found its way into a wide range of different application areas, for example signal processing [75], factor analysis [29, 44], system identification [77], chemometrics [20, 66] and more recently, general data mining [11, 58, 70] including image processing [17, 72] and pattern recognition [10, 47], as well as process monitoring and quality control [1, 82] including multiway [48], multiblock [52] and

multiscale [3] extensions. This success is mainly related to the ability of PCA to describe significant information/variation within the recorded data typically by the first few score variables, which simplifies data analysis tasks accordingly.

Sylvester [67] formulated the idea behind PCA, in his work the removal of redundancy in bilinear quantics, that are polynomial expressions where the sum of the exponents are of an order greater than 2, and Pearson [51] laid the conceptual basis for PCA by defining lines and planes in a multivariable space that present the closest fit to a given set of points. Hotelling [28] then refined this formulation to that used today. Numerically, PCA is closely related to an eigenvector-eigenvalue decomposition of a data covariance, or correlation matrix and numerical algorithms to obtain this decomposition include the iterative NIPALS algorithm [78], which was defined similarly by Fisher and MacKenzie earlier in [80], and the singular value decomposition. Good overviews concerning PCA are given in Mardia *et al.* [45], Joliffe [32], Wold *et al.* [80] and Jackson [30].

The aim of this article is to review and examine nonlinear extensions of PCA that have been proposed over the past two decades. This is an important research field, as the application of linear PCA to nonlinear data may be inadequate [49]. The first attempts to present nonlinear PCA extensions include a generalization, utilizing a nonmetric scaling, that produces a nonlinear optimization problem [42] and constructing a curves through a given cloud of points, referred to as principal curves [25]. Inspired by the fact that the reconstruction of the original variables, $\widehat{\mathbf{z}}$ is given by:

$$\widehat{\mathbf{z}} = \mathbf{P}\mathbf{t} = \overbrace{\mathbf{P}\underbrace{\left(\mathbf{P}^T\mathbf{z}\right)}_{\text{mapping}}}^{\text{demapping}}, \tag{1.2}$$

that includes the determination of the score variables (mapping stage) and the determination of $\widehat{\mathbf{z}}$ (demapping stage), Kramer [37] proposed an *autoassociative neural network* (ANN) structure that defines the mapping and demapping stages by neural network layers. Tan and Mavrovouniotis [68] pointed out, however, that the 5 layers network topology of autoassociative neural networks may be difficult to train, i.e. network weights are difficult to determine if the number of layers increases [27].

To reduce the network complexity, Tan and Mavrovouniotis proposed an *input training* (IT) network topology, which omits the mapping layer. Thus, only a 3 layer network remains, where the reduced set of nonlinear principal components are obtained as part of the training procedure for establishing the IT network. Dong and McAvoy [16] introduced an alternative approach that divides the 5 layer autoassociative network topology into two 3 layer topologies, which, in turn, represent the nonlinear mapping and demapping functions. The output of the first network, that is the mapping layer, are the score variables which are determined using the principal curve approach.

The second layer then represents the demapping function for which the score variables are the inputs and the original variables are the outputs. Jia *et al.* [31] presented a critical review of the techniques in references [16, 68] and argued that the incorporation of a principal curve algorithm into a neural network structure [16] may only cover a limited class of nonlinear functions. Hence, the IT network topology [68] may provide a more effective nonlinear compression than the technique by Dong and McAvoy [16]. In addition, Jia *et al.* [31] further refined the IT concept by introducing a linear compression using PCA first, which is followed by the application of the IT algorithm using the scaled linear principal components.

More recently, *Kernel PCA* (KPCA) has been proposed by Schölkopf [56, 57]. KPCA first maps the original variable set $\mathbf{z}$ onto a high-dimensional feature space using the mapping function $\mathbf{\Phi}(\mathbf{z})$. Then, KPCA performs a conventional linear principal component analysis on $\mathbf{\Phi}(\mathbf{z})$. The KPCA approach takes advantage of the fact that the mapping function $\mathbf{z} \mapsto \mathbf{\Phi}(\mathbf{z})$ does not need to be known *a priori*. Furthermore, this mapping function can be approximated using Kernel functions in a similar fashion to a radial basis function neural network. In fact, the identification of a KPCA model utilizes scalar products of the observations, which are then nonlinearly transformed using Kernel functions. This presents a considerable advantage over neural network approaches since no nonlinear optimization procedure needs to be considered. Resulting from this conceptual simplicity and computational efficiency, KPCA has recently found its way into a wide range of applications, most notably in the areas of face recognition [36], image de-noising [40] and industrial process fault detection [12, 81].

This article is divided into the following sections. A brief review of PCA including its most important properties is given next, prior to the introduction of a nonlinearity test. Section 4 then details nonlinear extensions of PCA. Section 5 then critically evaluates existing work on NLPCA in terms of computational demand in computing a model as well as generalization issues and provides a roadmap for future research work.

1.2 PCA Preliminaries

PCA analyses a data matrix $\mathbf{Z} \in \mathbb{R}^{K \times N}$ that possesses the following structure:

$$\mathbf{Z} = \begin{bmatrix} z_{11} & z_{21} & z_{13} & \cdots & z_{1j} & \cdots & z_{1N} \\ z_{21} & z_{22} & z_{23} & \cdots & z_{2j} & \cdots & z_{2N} \\ z_{31} & z_{32} & z_{33} & \cdots & z_{3j} & \cdots & z_{3N} \\ \vdots & \vdots & \vdots & & \vdots & & \vdots \\ z_{i1} & z_{i2} & z_{i3} & \cdots & z_{ij} & \cdots & z_{iN} \\ \vdots & \vdots & \vdots & & \vdots & & \vdots \\ z_{K-1,1} & z_{K-1,2} & z_{K-1,3} & \cdots & z_{K-1,j} & \cdots & z_{K-1,N} \\ z_{K1} & z_{K2} & z_{K3} & \cdots & z_{Kj} & \cdots & z_{KN} \end{bmatrix}, \tag{1.3}$$

where N and K are the number of recorded variables and the number of available observations, respectively. Defining the rows and columns of $\mathbf{Z}$ by vectors $\mathbf{z}_i \in \mathbb{R}^N$ and $\boldsymbol{\zeta}_j \in \mathbb{R}^K$, respectively, $\mathbf{Z}$ can be rewritten as shown below:

$$\mathbf{Z} = \begin{bmatrix} \mathbf{z}_1^T \\ \mathbf{z}_2^T \\ \mathbf{z}_3^T \\ \vdots \\ \mathbf{z}_i^T \\ \vdots \\ \mathbf{z}_{K-1}^T \\ \mathbf{z}_K^T \end{bmatrix} = \begin{bmatrix} \boldsymbol{\zeta}_1 \; \boldsymbol{\zeta}_2 \; \boldsymbol{\zeta}_3 \cdots \boldsymbol{\zeta}_j \cdots \boldsymbol{\zeta}_N \end{bmatrix} . \tag{1.4}$$

The first and second order statistics of the original set variables $\mathbf{z}^T = \begin{pmatrix} z_1 \; z_2 \; z_3 \cdots \quad z_j \cdots z_N \end{pmatrix}$ are:

$$E\left\{\mathbf{z}\right\} = \mathbf{0} \qquad E\left\{\mathbf{z}\mathbf{z}^T\right\} = \mathbf{S}_{ZZ} \tag{1.5}$$

with the correlation matrix of $\mathbf{z}$ being defined as $\mathbf{R}_{ZZ}$.

The PCA analysis entails the determination of a set of score variables t_k, $k \in \left\{ 1 \; 2 \; 3 \cdots n \right\}$, $n \leq N$, by applying a linear transformation of $\mathbf{z}$:

$$t_k = \sum_{j=1}^{N} p_{kj} z_j \tag{1.6}$$

under the following constraint for the parameter vector

$$\mathbf{p}_k^T = \begin{pmatrix} p_{k1} \; p_{k2} \; p_{k3} \cdots p_{kj} \cdots p_{kN} \end{pmatrix} :$$

$$\sqrt{\sum_{j=1}^{N} p_{kj}^2} = \|\mathbf{p}_k\|_2 = 1 \; . \tag{1.7}$$

Storing the score variables in a vector $\mathbf{t}^T = \begin{pmatrix} t_1 \; t_2 \; t_3 \cdots t_j \cdots t_n \end{pmatrix}$, $\mathbf{t} \in \mathbb{R}^n$ has the following first and second order statistics:

$$E\left\{\mathbf{t}\right\} = \mathbf{0} \qquad E\left\{\mathbf{t}\mathbf{t}^T\right\} = \boldsymbol{\Lambda} \; , \tag{1.8}$$

where $\boldsymbol{\Lambda}$ is a diagonal matrix. An important property of PCA is that the variance of the score variables represent the following maximum:

$$\lambda_k = \arg\max_{\mathbf{p}_k} \left\{ E\left\{t_k^2\right\} \right\} = \arg\max_{\mathbf{p}_k} \left\{ E\left\{\mathbf{p}_k^T \mathbf{z}\mathbf{z}^T \mathbf{p}_k\right\} \right\} \; , \tag{1.9}$$

that is constraint by:

$$E\left\{\begin{pmatrix} t_1 \\ t_2 \\ t_3 \\ \vdots \\ t_{k-1} \end{pmatrix} t_k \right\} = \mathbf{0} \qquad \|\mathbf{p}_k\|_2^2 - 1 = 0\,. \tag{1.10}$$

Anderson [2] indicated that the formulation of the above constrained optimization can alternatively be written as:

$$\lambda_k = \arg\max_{\mathbf{p}} \left\{ E\left\{\mathbf{p}^T\mathbf{z}\mathbf{z}^T\mathbf{p}\right\} - \lambda_k\left(\mathbf{p}^T\mathbf{p} - 1\right)\right\} \tag{1.11}$$

under the assumption that λ_k is predetermined. Reformulating (1.11) to determine $\mathbf{p}_k$ gives rise to:

$$\mathbf{p}_k = \arg\frac{\partial}{\partial\mathbf{p}} \left\{ E\left\{\mathbf{p}^T\mathbf{z}\mathbf{z}^T\mathbf{p}\right\} - \lambda_k\left(\mathbf{p}^T\mathbf{p} - 1\right)\right\} = \mathbf{0} \tag{1.12}$$

and produces

$$\mathbf{p}_k = \arg\left\{ E\left\{\mathbf{z}\mathbf{z}^T\right\}\mathbf{p} - 2\lambda_k\mathbf{p}\right\} = \mathbf{0}\,. \tag{1.13}$$

Incorporating (1.5) allows constructing an analytical solution of this constrained optimization problem:

$$\left[\mathbf{S}_{ZZ} - \lambda_k\mathbf{I}\right]\mathbf{p}_k = \mathbf{0}\,, \tag{1.14}$$

which implies that the kth largest eigenvalue of $\mathbf{S}_{ZZ}$ is the variance of the kth score variable and the parameter vector $\mathbf{p}_k$, associated with λ_k, stores the kth set of coefficients to obtain the kth linear transformation of the original variable set $\mathbf{z}$ to produce t_k. Furthermore, given that $\mathbf{S}_{ZZ}$ is a positive definite or semidefinite matrix it follows that the eigenvalues are positive and real and the eigenvectors are mutually orthonormal. The solution of Equation (1.14) also implies that the score variables are statistically independent, as defined in (1.10), which follows from:

$$\widehat{\mathbf{S}}_{ZZ} = \frac{1}{K-1}\widehat{\mathbf{Z}}^T\widehat{\mathbf{Z}} = \widehat{\mathbf{P}}\widehat{\mathbf{\Lambda}}\widehat{\mathbf{P}}^T \Longrightarrow \frac{1}{K-1}\widehat{\mathbf{P}}^T\mathbf{Z}^T\mathbf{Z}\widehat{\mathbf{P}} = \frac{1}{K-1}\widehat{\mathbf{T}}^T\widehat{\mathbf{T}} = \widehat{\mathbf{\Lambda}}\,. \tag{1.15}$$

Here, the index $\widehat{\circ}$ represents estimates of the covariance matrix, its eigenvectors and eigenvalues and the score matrix using the reference data stored in $\mathbf{Z}$. A solution of Equations (1.9) and (1.10) can be obtained using a singular value decomposition of the data covariance matrix $\widehat{\mathbf{S}}_{ZZ}$ or the iterative Power method [22].

1.3 Nonlinearity Test for PCA Models

This section discusses how to determine whether the underlying structure within the recorded data is linear or nonlinear. Kruger *et al.* [38] introduced this nonlinearity test using the principle outlined in Fig. 1.1. The left plot in this figure shows that the first principal component describes the underlying linear relationship between the two variables, z_1 and z_2, while the right plot describes some basic nonlinear function, indicated by the curve.

By dividing the operating region into several disjunct regions, where the first region is centered around the origin of the coordinate system, a PCA model can be obtained from the data of each of these disjunct regions. With respect to Fig. 1.1, this would produce a total of 3 PCA models for each disjunct region in both cases, the linear (left plot) and the nonlinear case (right plot). To determine whether a linear or nonlinear variable interrelationship can be extracted from the data, the principle idea is to take advantage of the residual variance in each of the regions. More precisely, *accuracy bounds* that are based on the residual variance are obtained for *one of the PCA models*, for example that of disjunct region I, and the residual variance of the *remaining PCA models* (for disjunct regions II and III) are benchmarked against *these bounds*. The test is completed if *each of the PCA models* has been used to determine accuracy bounds which are then benchmarked against the residual variance of the respective *remaining PCA models*.

The reason of using the residual variance instead of the variance of the retained score variables is as follows. The residual variance is independent of the region if the underlying interrelationship between the original variables is linear, which the left plot in Fig. 1.1 indicates. In contrast, observations that have a larger distance from the origin of the coordinate system will, by default, produce a larger projection distance from the origin, that is a larger score value. In this respect, observations that are associated with an

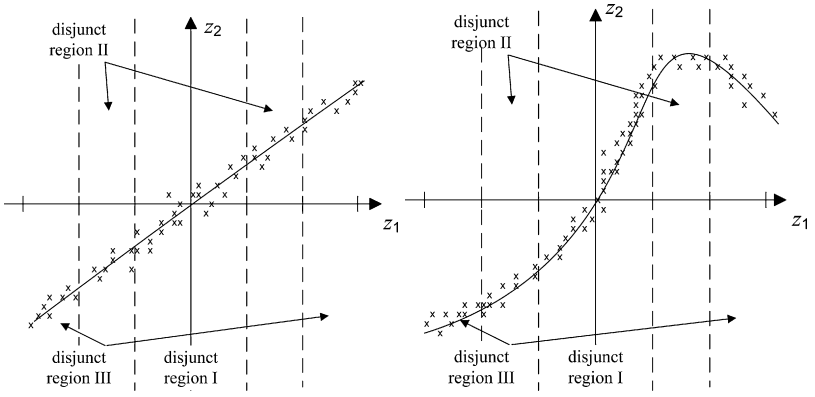

Fig. 1.1. Principle of nonlinearity test

adjunct region that are further outside will logically produce a larger variance irrespective of whether the variable interrelationships are linear or nonlinear.

The detailed presentation of the nonlinearity test in the remainder of this section is structured as follows. Next, the assumptions imposed on the nonlinearity test are shown, prior to a detailed discussion into the construction of disjunct regions. Subsection 3.3 then shows how to obtain statistical confidence limits for the nondiagonal elements of the correlation matrix. This is followed by the definition of the accuracy bounds. Finally, a summary of the nonlinearity test is presented and some example studies are presented to demonstrate the working of this test.

1.3.1 Assumptions

The assumptions imposed on the nonlinearity test are summarized below [38].

1. The variables are mean-centered and scaled to unit variance with respect to disjunct regions for which the accuracy bounds are to be determined.
2. Each disjunct region has the same number of observations.
3. A PCA model is determined for one region where the the accuracy bounds describe the variation for the sum of the discarded eigenvalues in that region.
4. PCA models are determined for the remaining disjunct regions.
5. The PCA models for each region include the same number of retained principal components.

1.3.2 Disjunct Regions

Here, we investigate how to construct the disjunct regions and how many disjunct regions should be considered. In essence, dividing the operating range into the disjunct regions can be carried out through prior knowledge of the process or by directly analyzing the recorded data. Utilizing *a priori* knowledge into the construction of the disjunct regions, for example, entails the incorporation of knowledge about distinct operating regions of the process. A direct analysis, on the other hand, by applying scatter plots of the first few retained principal components could reveal patterns that are indicative of distinct operating conditions. Wold *et al.* [80], page 46, presented an example of this based on a set of 20 "natural" amino acids.

If the above analysis does not yield any distinctive features, however, the original operating region could be divided into two disjunct regions initially. The nonlinearity test can then be applied to these two initial disjunct regions. Then, the number of regions can be increased incrementally, followed by a subsequent application of the test. It should be noted, however, that increasing the number of disjunct regions is accompanied by a reduction in the number of observations in each region. As outlined the next subsection, a sufficient number of observations are required in order to prevent large Type I and II

errors for testing the hypothesis of *using a linear model* against the alternative hypothesis of *rejecting that a linear model can be used*.

Next, we discuss which of the disjunct regions should be used to establish the accuracy bounds. Intuitively, one could consider the most centered region for this purpose or alternatively, a region that is at the margin of the original operating region. More practically, the region at which the process is known to operate most often could be selected. This, however, would require *a priori* knowledge of the process. However, a simpler approach relies on the incorporation of the cross-validation principle [64, 65] to automate this selection. In relation to PCA, cross-validation has been proposed as a technique to determine the number of retained principal components by Wold [79] and Krzanowski [39].

Applied to the nonlinearity test, the cross-validation principle could be applied in the following manner. First, select one disjunct region and compute the accuracy bounds of that region. Then, benchmark the residual variance of the remaining PCA models against this set of bounds. The test is completed if accuracy bounds have been computed for each of the disjunct regions and the residual variances of the PCA models of the respective remaining disjunct regions have been benchmarked against these accuracy bounds. For example, if 3 disjunct regions are established, the PCA model of the first region is used to calculate accuracy bounds and the residual variances of the 3 PCA models (one for each region) is benchmarked against this set of bounds. Then, the PCA model for the second region is used to determine accuracy bounds and again, the residual variances of the 3 PCA models are benchmarked against the second set of bounds. Finally, accuracy bounds for the PCA model of the 3rd region are constructed and each residual variance is compared to this 3rd set of bounds. It is important to note that the PCA models will vary depending upon which region is currently used to compute accuracy bounds. This is a result of the normalization procedure, since the mean and variance of each variable may change from region to region.

1.3.3 Confidence Limits for Correlation Matrix

The data correlation matrix, which is symmetric and positive semidefinite, for a given set of N variables has the following structure:

$$\mathbf{R}_{ZZ} = \begin{bmatrix} 1 & r_{12} & \cdots & r_{1N} \\ r_{21} & 1 & \cdots & r_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ r_{N1} & r_{N2} & \cdots & 1 \end{bmatrix} . \tag{1.16}$$

Given that the total number of disjunct regions is m the number of observations used to construct any correlation matrix is $\widetilde{K} = K/m$, rounded to the nearest integer. Furthermore, the correlation matrix for constructing the