Nonlinear dimensionality reduction of gene expression data

Nilsson, Jens

2006

Nonlinear Dimensionality Reduction of Gene Expression Data

Jens Nilsson

Lund University

Faculty of Engineering
Centre for Mathematical Sciences
Mathematics

Mathematics
Centre for Mathematical Sciences
Lund University
Box 118
SE-221 00 Lund
Sweden

http://www.maths.lth.se/

Licentiate Theses in Mathematical Sciences 2006:1
ISSN 1404-028X

ISBN 91-631-8376-5, 978-91-631-8376-8
LUTFMA-2021-2006

Printed in Sweden by KFS, Lund 2006

Preface

This thesis deals with manifold learning and nonlinear dimensionality reduction in gene expression data analysis. It contains experimental evaluation studies and methodological development within the field of manifold learning.

The work has been conducted at the Centre for Mathematical Sciences, Faculty of Engineering, Lund University, in a PhD project which is part of the Industrial PhD program in Medical Bioinformatics and financed partly by AstraZeneca and the Swedish Knowledge Foundation.

The thesis is based on the following two papers:

> J. Nilsson, T. Fioretos, M. Höglund and M. Fontes, Approximate geodesic distances reveal biologically relevant structures in microarray data. *Bioinformatics* 20(6): 874–880, 2004
>
> J. Nilsson and F. Andersson, Circuit models for manifold learning. *submitted*, 2006

where the last paper is a development of

> F. Andersson and J. Nilsson, Nonlinear dimensionality reduction using circuit models. *Proc. 14th Scandinavian Conf. on Image Analysis*, Springer-Verlag, 2005.

Outside the thesis, contributions have been made to

> A. Andersson, P. Edén, D. Lindgren, J. Nilsson, C. Lassen, J. Heldrup, M. Fontes, Å. Borg, F. Mitelman, B. Johansson, M. Höglund, T. Fioretos, Gene expression profiling of leukemic cell lines reveals conserved molecular signatures among subtypes with specific genetic aberrations. *Leukemia* 19(6):1042-50, 2005.

Acknowledgements

Many people deserve credits for making this journey both easier and more enjoyable. First, I would like to thank my supervisor Magnus Fontes and co-supervisors Thoas Fioretos, Per Broberg and Yudi Pawitan for providing competent and enthusiastic guidance in a friendly atmosphere.

The colleagues and friends at the Centre for Mathematical Sciences provide good company and a nice atmosphere daily. In particular, I would like to mention Fredrik Andersson, Erik Alpkvist, Henrik Bengtsson, Martin Dahlgren and Azra Kurbasic. Thank you for your support and for good collaborations. Johan Råde has made valuable contributions which have been of great benefit. Kalle Åström played an important role in the initial stages of the project by pointing at Isomap as a possible tool for microarray data analysis. Finally, the department soccer team should not be forgotten. Practise makes perfect!

I would further like to thank AstraZeneca R&D Lund and the colleagues at the Disease Association 2 team and the department of Biological Sciences there. Per Broberg is a valued and constant support.

I have also had the opportunity to collaborate with the department of Clinical Genetics at Lund university hospital. I have indeed learned a lot from Thoas Fioretos, Mattias Höglund, Anna Andersson and their colleagues there.

Thanks also to Per-Erik Jansson at the Centre for Medical Innovations (CMI), the Karolinska Institute and all students and other people involved in the Research school in Medical Bioinformatics.

Regarding this thesis, I have received valuable comments and help with proof-reading from Magnus Fontes, Per Broberg, Henrik Bengtsson, Thoas Fioretos, Erik Alpkvist and Fredrik Andersson. For this I am grateful. Magnus has made important contributions to Chapter 3, and Henrik, likewise, to Chapter 2.

Finally, I would like to thank Jan, Lena, Ida, Lola and Siham for all their love and support. Thanks also to all friends and family not mentioned in the above.

Lund, January 2006,
Jens Nilsson

Contents

Chapter 1

Introduction

1.1 Motivation

What causes the difference between a nerve cell and a muscle cell? And what sets these apart from, say, a red blood cell or a skin cell? After all, all cells in our body carry, in principle, the same genetic code, the blueprints of all molecules the cell can manufacture. The answer is that different genes are activated in different cell types. But then we may still wonder, what causes this differentiation in gene expression, when at one point in time we start as a single cell? And what lies behind a living cells impressive ability to answer external and internal stimuli with appropriate reactions? It seems that these are emergent properties of the cells genetic regulatory system, that is, the complex network of chemical reactions, through which the activities of the genes are linked.

Previously, science has been limited to the study of these systems from outside, by registering only emergent, high-level properties, or at best measuring the activity of a few genes at the time. However, recent advances in molecular biology and biotechnology have made it possible to monitor the activity levels of thousands of genes simultaneously. Using microarray technology, it is possible to measure the amount of mRNA molecules stemming from, in principle, each one of all the genes in the genome at a given moment. In other words, we now have means to sample the overall state[1] of the cellular genetic regulatory network under different conditions. Such data have great potential value. It can, for example, be used to infer the functional role of a given gene, to diagnose cell samples or to decipher complex disease mechanisms.

The size and complexity of the system from which data is collected requires the use of computational methods in order to extract meaningful patterns. Approaches stemming from the traditions of mathematics, statistics, physics and computer science have all proven useful in the analysis of gene expression data.

We may represent microarray samples as points in a gene expression space, with coordinates describing the state of the regulatory system. In this way, a given number of cell samples defines a point cloud in gene expression space. Motivated by the assumption that functional relations between genes in the regulatory networks are, to a substantial extent,

[1] At least on the level of mRNA molecules.

nonlinear, we make the assumption that samples lie on a, possibly nonlinear, Riemannian manifold in gene expression space. Furthermore, we assume that the metric on this manifold carries biologically relevant information.

One approach to the problem of pattern extraction from gene expression data is to apply some dimensionality reduction method. These have in common that they attempt to map the data onto a point configuration in lower-dimensional space that optimally represents some property of the data, typically geometrical or statistical.

Under the assumption that data are samples from a general Riemannian manifold, standard, linear methods for dimensional reduction, such as principal component analysis are not optimal choices, since they fail to recover the intrinsic data geometry. A significant trend in machine learning during recent years has been the development of nonlinear dimensionality reduction methods, that is, methods that take into account that data in high-dimensional input space may lie on manifolds with an intrinsic geometry different from the surrounding space. More concretely, these methods allow data to be sampled from 'curved surfaces' in input space. Nonlinear dimensionality reduction *per se* is nothing new, but the methods referred to above have in common that they rely on the spectral decomposition of different specially constructed matrices reflecting some geometrical property of the data. In that way, computationally demanding optimization techniques who risk yielding non-optimal solutions are avoided.

The focus of this work is on the use of nonlinear dimensionality reduction methods to represent the point cloud of gene expression measurements in a lower-dimensional space, for example, for the purpose of visualizing underlying patterns in the data. The aim is to show that it is important and beneficial to take nonlinear structures in the data into account. Moreover, work is done within the methodological development of nonlinear dimensionality reduction and manifold learning. More specifically, a method for robust estimation of geodesic distances is proposed.

1.2 Organization of the thesis

The next chapter gives an introduction to the biological, technological and computational issues involved in gene expression data analysis and puts them in a mathematical framework. In Section 2.1 the biology of gene regulation in the cell is described. Section 2.2 introduces the microarray techniques, with which gene expression levels can be measured. The series of data processing steps that a microarray sample goes through, from the raw image, as produced by the microarray scanner, to the final representation of the data, which serves as the basis for biological conclusions, is reviewed in Section 2.3. Finally, in Section 2.4, the view of microarray data as samples from a manifold in expression space is discussed in more detail.

Chapter 3 provides a review of the field of dimensionality reduction, with a special emphasis on spectral methods. Section 3.1 gives an introduction to the dimensionality reduction problem, aimed to a non-mathematical audience. Following this, definitions of the manifold learning and nonlinear dimensionality reduction problems together with related concepts are given in Section 3.2. Sections 3.3 to 3.5 cover, in turn, the methods of Principal Component Analysis (PCA), Kernel PCA and Multidimensional Scaling (MDS). After this, we turn to graph-based methods for spectral dimensionality reduction

and present Isomap (Section 3.6), Laplacian Eigenmaps (Section 3.7) and Locally Linear Embedding (Section 3.8). The chapter is concluded in Section 3.9 by a discussion on how the covered spectral methods can be described within the kernel framework.

Chapter 4 consists of the article *Approximate geodesic distances reveal biologically relevant structures in microarray data* [Nilsson et al., 2004], where Isomap is applied to compute approximate geodesic distances for two microarray data sets. We show that the use of approximate geodesic distances as a dissimilarity measure, compared to the standard Euclidean metric, can yield lower-dimensional visualizations of the data in which diagnostic sample groups appear significantly more clearly. This enables us to make a detailed biological interpretation of the obtained lower-dimensional representations. We conclude that the results show the benefit and importance of taking nonlinearities into account in gene expression data analysis, and in particular in dimensionality reduction.

Chapter 5 consists of the paper *Circuit models for manifold learning* [Nilsson and Andersson, 2006], where a method for robust estimation of geodesic distances is presented. The Isomap algorithm has the disadvantage that it is topologically unstable, that is, small perturbations on the input data may result in large changes in the approximate geodesic distances. This is because shortcuts may appear in the adjacency graph, connecting geodesically distant domains of the manifold. By interpreting the adjacency graph as an electric circuit we define a distance measure based on the propagation of charges in this circuit. By analyzing three different data sets we show that dimensionality reduction based on these distances is more topologically stable than the corresponding Isomap results. We also compare with Laplacian Eigenmaps and PCA and discuss the influence of algorithm parameter choices on the performance.

Chapter 6 concludes the thesis with a discussion on the causes and the meaning of the results in Chapter 4 and 5. Some additional results are presented for the sake of completeness. Finally, some important directions of future work are pointed out.

Chapter 2

Gene expression data

2.1 Cellular biology and gene regulation

The cell is a complex machinery — a collection of organic molecules, intricately interacting, constituting what we may define as the basic unit of life. This microscopic mixture of molecules possesses an extraordinary ability to communicate with its environment and to regulate its own state according to internal and external stimuli.

The structural and functional building blocks of the cell are primarily *proteins*, but also *ribonucleic acids (RNA)*. Many of these molecules cater reactions such as signalling and metabolism while others make up the skeleton and shell of the cell thus defining it spatially in its environment. The cell manufactures these molecules itself, relying on molecular blueprints that are inherited from cell to cell. The blueprints are stored in the *deoxyribonucleic acid (DNA)*, a molecule shaped as a double helix, where the two strands are joined together through pairs of *nucleotides*. It is the sequence of nucleotides that determines the information content in the DNA. The set of nucleotides in the DNA is made up of *adenine (A)*, *thymine (T)*, *guanine (G)* and *cytosine (C)*, so the information is encoded in a four-letter alphabet. The two strands of the double helix are complementary to each other since A always pairs with T and G always pairs with C. A *gene* is a portion of the DNA which contains the instructions for a specific molecule, its *gene product*.

In principle, all cells in a given multicellular organism carry the same genetic code, identical to the one of the original fertilized egg. Nevertheless, higher order species consist of highly specialized cell types, appearing in different locations of the body, having different tasks. So why do skin cells, nerve cells and blood cells, who all have the same genetic code, behave so widely different? The answer is that different genes are active, or *expressed* in the different cell types, making them produce their own specific set of molecules.

The *protein synthesis*, that is, the process of producing a protein from the information in its corresponding gene can be divided into two phases — *transcription* and *translation* (Figure 2.1).

During transcription, the genetic code of the gene is copied to a *messenger RNA (mRNA)* molecule, a single-stranded nucleic acid carrying the same nucleotides as DNA with the exception of thymine whose role is instead taken by *uracil (U)*. Transcription

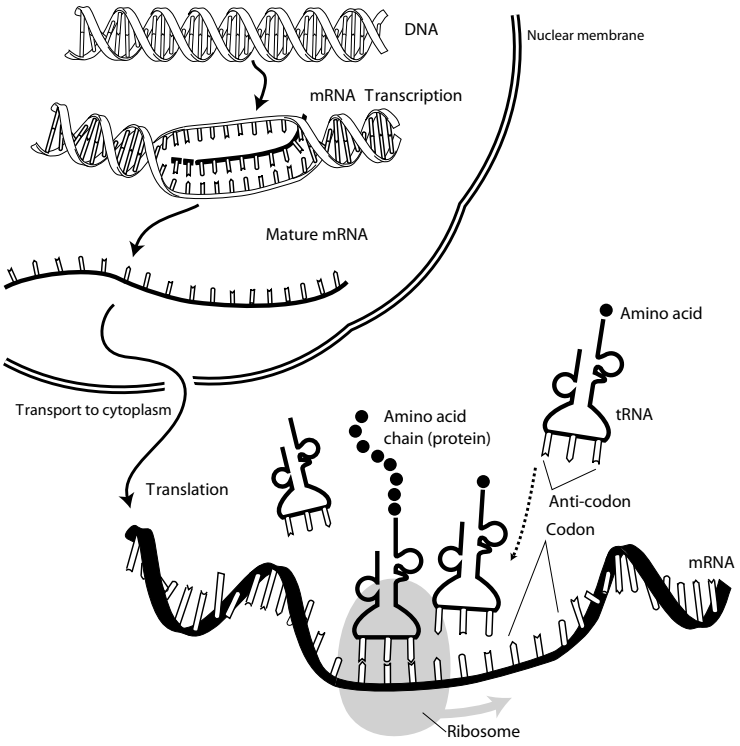

Figure 2.1: A schematic illustration of the cells protein synthesis. The figure is printed by courtesy of the National Human Genome Research Institute, the National Institutes of Health.

starts when the protein RNA polymerase binds to the *promoter region*, the start of the gene, and locally unzips the DNA helix so that the strands become free for reading. The RNA polymerase propagates along the strand while constructing an mRNA molecule by adding nucleotides complementary to those being passed by on the DNA molecule. Eventually, the RNA polymerase reaches a *terminator region* and stops transcribing, whereby the mRNA is released and the DNA resumes its double helix configuration. Following this, the primary mRNA is processed into mature mRNA by other molecules, for example by removing the parts corresponding to *introns*, non-coding regions of the DNA, in a process called *splicing*.

Following transcription, translation takes place, where the four-letter alphabet of the DNA and mRNA is translated into the alphabet of proteins. Like the nucleic acids, proteins are polymers, albeit consisting of sequences of *amino acids* instead of nucleotides. The number of amino acids is 20 so the protein alphabet is one of 20 letters. In order to represent 20 amino acids with four nucleotides we need three nucleotides per amino acid. Such a three-nucleotide word is denoted a *codon*.[1] The actual translation between the two

[1] Since $4^3 = 64 > 20$ there is a degeneracy in this representation; some amino acid are coded for by more

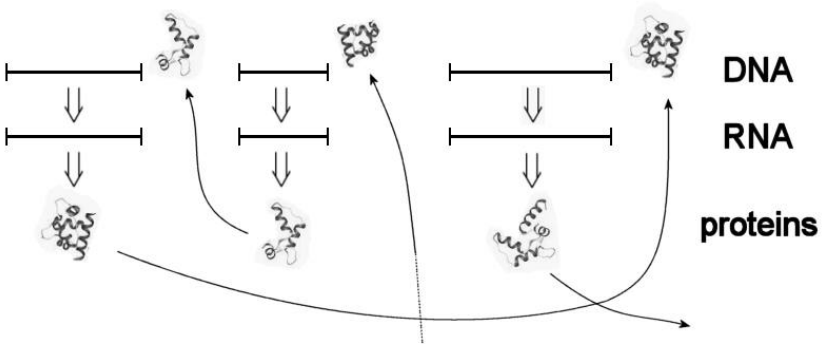

Figure 2.2: A schematic illustration of a small subset of a gene regulatory network. Gene products influence the expression of other genes.

alphabets is accomplished by *transfer RNA* molecules *(tRNA)* which attach themselves to the mRNA. The tRNA has one end with a specific *anticodon*, that is, a complementary codon, and another end to which the corresponding amino acid is attached. The last step in the translation is performed by the ribosomes which join the sequence of amino acids found on the tRNA along the mRNA together to form the protein.

The same protein can attain different spatial shapes in the three-dimensional space, constituting a number of *folds* of the protein, each having their own properties. For this, and other reasons, it is convenient to define the *expression level* of a gene as the amount of mRNA in the cell transcribed from it at a given instant. We also define the *expression profile* of a cell as the collected expression levels of all genes.

The production of RNA and proteins from a given gene does not take place independently of the expression of other genes. Conversely, gene products influence the production of other gene products using positive or negative feedback (Figure 2.2). This regulation is essential for the cell to be able to respond to internal and external circumstances and takes place on all levels in the chain of reactions that produce a protein from a gene sequence.

To enable transcription of a gene, the binding of certain proteins, *transcription factors*, to the DNA, is necessary. Different genes are either activated or repressed by different combinations of one or several transcription factors. Transcriptional control is not the only means of regulation. After transcription, mRNA molecules may interact with other gene products, resulting in altered structure or lifetime of the mRNA. After translation, subsequent protein-protein reactions may be required to finalize the functional protein.

The description above only sketches a few of the ways that genes interact to regulate each others expression. The main conclusion is that the cell can be viewed as a large dynamical system with different molecules interacting with each other. The explicit study of such *genetic regulatory networks* is often referred to as *systems biology* (although other wider definitions of the term are also used). Mathematical modelling of genetic regula-

than one codon. On the other hand, one codon codes for at most one amino acid.

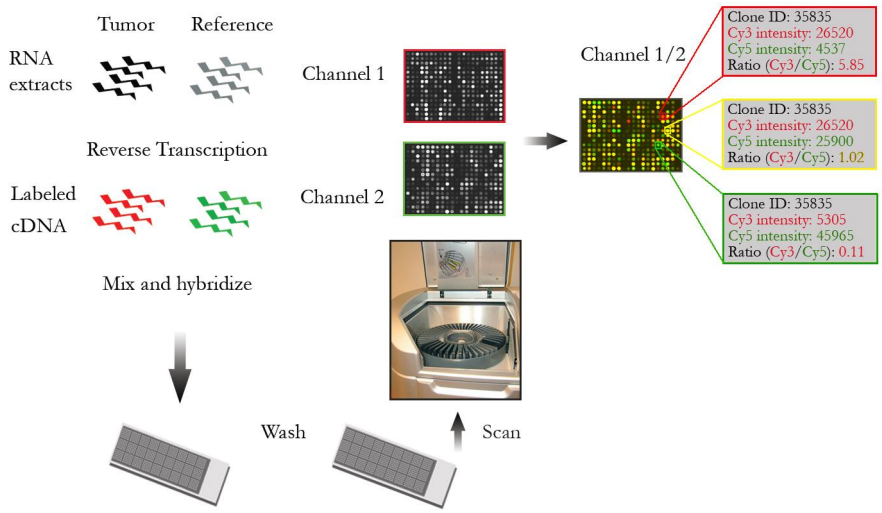

Figure 2.3: The spotted microarray technique. The figure is printed by courtesy of Anna Andersson, dept. of Clinical Genetics, Lund University hospital.

tory networks had its principal breakthrough in the 1960's and various models have been proposed, ranging in complexity from discrete cellular automata models to detailed probabilistic models; see [de Jong, 2002] for a review.

2.2 Techniques for measuring gene expression

Enabled by advances in measurement technology and the sequencing of genomes, such as the human, in the Human Genome Project, the 1990's saw the emergence of technologies for global measurement of gene expression. Earlier techniques were limited to the study of a few genes at the time, while the *microarray* techniques gave biologists the tools to sample the expression of, in principle, the whole genome in one single measurement.

Microarrays measure the abundance of mRNA from the set of genes at a given moment. From a cell sample of interest, mRNA is extracted and put in contact with an array on which *probes*; complementary sequences (or subsequences) of the genes, have been attached. The different mRNA in the solution then bind to their corresponding complements on the chip, and the amount of mRNA for each gene can be optically measured by a laser scanner.

There are two main microarray platforms currently in use; *spotted microarrays* [Schena et al., 1995a,b] and *high-density synthetic oligonucleotide microarrays* [Lockhart et al., 1996]. These are basically two variations of the same general solution described above.

A spotted microarray[2] (Figure 2.3) has probes consisting of cDNA or long oligo strands

[2]The well known cDNA microarray technique falls under the category of spotted microarrays.

attached spot-wise on a glass slide in a grid shaped pattern. The platform is, in its most common form, a *two-channel technique*, meaning that in each measurement, the expression profiles of two cell samples are measured simultaneously.[3] After extracting RNA from the two samples it is reverse-transcribed to cDNA, and fluorescently labelled with Cy3 (green) for one sample and Cy5 (red) for the other. The cDNA molecules of the samples are denoted *targets*. After labelling, the two samples are mixed and put in contact with the probes on the slide. During *hybridization*, the targets bind to their corresponding probes, thus geometrically sorting the targets on the slide. Finally, each spot is illuminated by a laser at two different wavelengths; one yielding Cy3 fluorescence and one yielding Cy5 fluorescence. Thus two images are obtained; one with green spots and one with red spots, measuring the abundances of the respective sample targets. These images then goes through a number of image processing steps. First, the the spots need to be located and segmented out from the background. Second, the spot intensity and local background intensity is estimated from the pixels, commonly by taking the mean or the median of the pixel values. Thus for each spot, estimates of the red and green foreground and background intensities are available. For each channel, the expression level is estimated as

$$Y_i = FG_i - BG_i, \tag{2.1}$$

where FG_i and BG_i are the foreground and background estimates at spot j for the particular channel. In principle, the two channels could be treated separately, but in multiarray experiments it is common practise to use a *reference sample*, common to all arrays, in one of the channels. The expression levels of the other sample, the *query sample*, are then reported as relative values compared to the reference expressions, i.e.,

$$Y_i = \frac{Y_i^{\text{query}}}{Y_i^{\text{reference}}}.$$

Most commonly this ratio is subsequently transformed by taking the logarithm, as discussed in Section 2.3.2.

High-density synthetic oligonucleotide microarrays have a slightly different construction. Here we describe the widely spread Affymetrix$^{\text{TM}}$platform. The probes are made up of excerpts of gene sequences, with a typical length of 25 nucleotides, and probes for one gene is spread over the chip in order to decrease the influence of systematic spatial errors. Moreover, associated to each probe is a *mismatch probe*, where one nucleotide has been replaced by its complement, thus providing a means of estimating the amount of non-specific, or false positive binding. The mismatch probe and the *perfect match probe* constitute a *probe pair* and typical arrays hold 16-20 probe pairs per gene. As opposed to spotted microarrays, high-density oligonucleotide microarrays are single-channel, measuring one sample on each array. To prepare the target, mRNA is extracted from the cell and fluorescently labelled while converted to complementary RNA. The targets are hybridized to the chip and an image is generated using a laser scanner (Figure 2.4). An image from an oligonucleotide array is slightly more standardized than a spotted microarray image in terms of location and size of the probes on the image, but basically the same image

[3] However, both one- and multi-channel spotted microarray platforms exist.

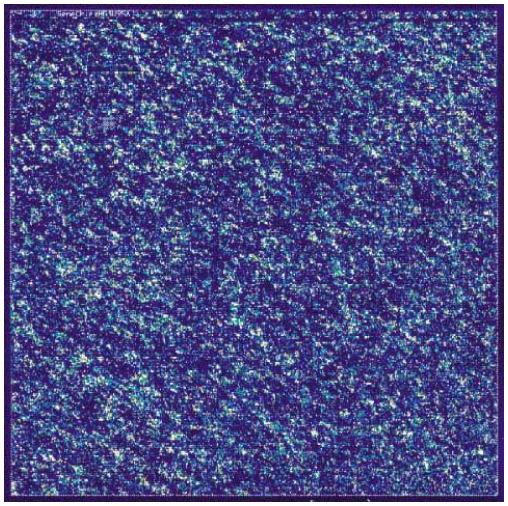

Figure 2.4: A scanned image from a high-density synthetic oligonucleotide microarray. The figure is printed by courtesy of Affymetrix.

processing steps as for spotted microarrays need to be performed — localization, segmentation and intensity estimation. Let PM_{ik} and MM_{ik} be the extracted perfect match and mismatch intensities for probe pair k of gene i, where $i = 1, \ldots, n$ and $k = 1, \ldots, K$, with K the number of probe pairs. For each probe pair we may, in the spirit of (2.1), estimate the expression level as

$$Y_{ik} = PM_{ik} - MM_{ik}.$$

Some approaches, however, disregard the mismatch intensities altogether and let $Y_{ik} = PM_{ik}$. The estimation of the expression level of a particular gene requires the summary of the probe pair expressions Y_{ik}, $k = 1, \ldots, K$ in one single value — an *expression index* as it is termed for oligonucleotide arrays. A straightforward way to do this is to compute the average, that is, $Y_i = \sum_k Y_{ik}/K$. However, this is sensitive to outliers, so instead a trimmed mean can be computed as in the *average difference*, which is defined as

$$Y_i = \sum_{k=1}^{K} w_{ik} Y_{ik}, \quad w_{ik} = \begin{cases} 1/\#A, & \text{if } k \in A; \\ 0, & \text{otherwise,} \end{cases}$$

where A is the set of probe pairs such that Y_{ik} is within three standard deviations from the mean. It has been shown that the average difference and similar measures are not optimal [Irizarry et al., 2003], why other, more sophisticated, expression indexes have been proposed. In multi-array experiments it becomes possible to take into account varying physical properties of the individual probes by taking advantage of the repeated experiments. One example is the *model-based expression index (MBEI)* [Li and Wong, 2001a]. Consider gene i on array j. Dropping the gene index, the MBEI is defined as the maximum likelihood estimate of ϑ_j in the model

$$Y_{jk} = \vartheta_j \varphi_k + \varepsilon_{jk},$$

where φ_k are probe specific affinities and ε_{jk} independent normally distributed errors. For each gene, the set of samples defines a point cloud in the space spanned by the probe pairs. The computation of the MBEI is equivalent to fitting a line with zero intercept through this cloud and letting ϑ_j be the length of the orthogonal projection of $[Y_{j1}, \ldots, Y_{jK}]$ onto this line. The MBEI can also be seen as a weighted average, where the weights are determined by the probe affinities, that is, the coefficients of the fitted line. Conceptually, for each probe set we may write

$$\vartheta_{jk} = \frac{Y_{jk} - \varepsilon_{jk}}{\varphi_k}.$$

The estimate of ϑ_j then becomes a weighted average with weights $w_k = \varphi_k^{-1}$. A similar, currently widely adopted expression index is the *robust multiarray average (RMA)* [Irizarry et al., 2003] which, for a given gene on array j, is defined as the estimates of μ_j in the model,

$$\log Y_{jk} = \mu_j + \varphi_k + \varepsilon_{jk},$$

where $\log Y_{jk}$ is the background-adjusted, normalized (see 2.3.2) and log-transformed PM intensity of probe pair k of the gene on array j; φ_k is a probe affinity and ε_{jk} an error term. Note that the RMA index is a log-scaled expression index.

Comparing the two platforms, spotted microarrays have the advantage that they are more flexible in the respect that they can be designed in the lab in terms of which probes to attach to the chip. On the other hand oligonucleotide chips have less risk of cross-hybridization and have a wider dynamical range, that is, the range at which the signal is linearly related to the mRNA abundance [Sebastiani et al., 2003].

2.3 Gene expression data analysis

The data processing, from scanned array images to the final biological interpretation involves a long series of computational manipulations and analyses of the data, each one, in their own respect, more or less challenging. We have already, in the previous section, seen the initial steps — the estimation of expression levels from the raw image data through spot identification, segmentation, intensity estimation and the computation of expression indexes. After this follows a number of *preprocessing* steps, where various transformations of the data is applied in order to filter out non-biological variation and to 'clean up' data to facilitate for subsequent analyses. At this stage, data is presumably ready to be analyzed in search for a biological interpretation. A range of *high-level analysis* methods exist that have as a common aim to extract biologically relevant patterns and information from the data. Clustering, classification, dimensionality reduction and other types of methods are all frequently applied in gene expression data analysis. Finally, the extracted structure needs validation, and here too, computational methods are helpful, for example while associating the results to prior knowledge which is often stored in large databases. Figure 2.5 gives an overview of the process of gene expression data analysis.

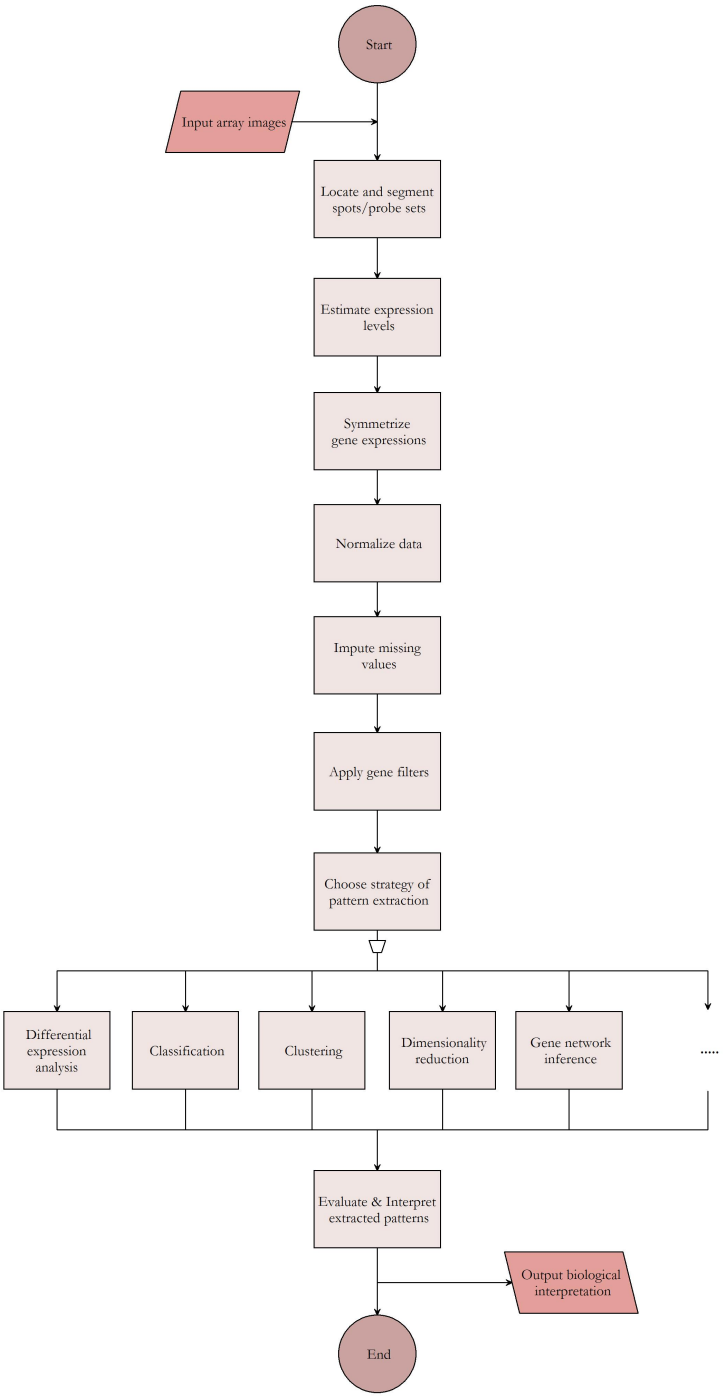

Figure 2.5: Flow chart of gene expression data analysis.

2.3.1 Preprocessing

Data transformations

As described in Section 2.2, expression values on a spotted microarray are computed as the logarithm of the ratio between the two channel expressions. The reason for taking the logarithm is to symmetrize between up- and down-regulation. In the original scale, down-regulation, that is when the query target is less abundant than the reference target, is squeezed in the interval $[0, 1]$, while up-regulation is spread over the interval $[1, \infty)$. Taking the logarithm makes up- and down-regulation symmetric in the interval $(-\infty, +\infty)$.

Normalization

Any given set of microarray measurements contains variation originating from different sources. The expression levels may vary across samples due to differences in the quantity of mRNA, different sample processing, scanner calibration, etc. Naturally, it is of interest to remove technical and experimental variation so that what remains is the biological variation, relevant to the study. This is the objective of *normalization*.

The sources of variation are many and all are not very well understood, why it is difficult to model them explicitly. Instead, typically some general assumption about invariance of certain quantities over samples is made. For example, in *total-intensity normalization* it is assumed that the true average gene expression is constant across samples, in which case each array is scaled by its total estimated expression. An extension of this idea is used in *quantile normalization* [Bolstad et al., 2003], where the distribution of expression values is assumed to be constant across samples. In this case, estimated expression values are transformed so that in a multidimensional quantile-quantile plot of the sample distributions, the quantiles lie along the main diagonal.

An alternative approach is to assume that a large majority of genes are non-differentially expressed across samples. Consider two microarray samples, for example the query and reference samples on a single spotted microarray or two samples hybridized on separate oligonucleotide arrays. In a scatter plot of gene expression in sample one against sample two, the points should lie along the main diagonal under the assumption that most genes are non-differentially expressed. If, in the observed data, they do not, the actual relation between the samples can be estimated by fitting a line through the point cloud. The data can then be transformed so that this line lies along the main diagonal. In in its most simple form this methodology rotates the point cloud $45°$ and applies linear regression, but more commonly the nonlinear *lowess regression* is used [Yang et al., 2002]. Generalizations for sets of more than two samples can be found in [Åstrand, 2003, Bolstad et al., 2003].

Under some circumstances none of the assumptions underlying the normalization methods described above are valid. This is, for example, the case if the microarray contains relatively few probes, the majority of which are known to be involved in the biological process under study. In this case, normalization is often based on the assumption that expression properties, like those described above, of a subset of the genes is invariant. This subset can be genes that are biologically known to have a constant expression, so called *housekeeping genes*, or it can be so called *spike-in genes* from some other organism whose mRNA is added in known amounts early in the experimental process. If no prior knowledge about invariant genes is at hand, a suitable subset can be selected using for example

the invariant set algorithm by Li and Wong [2001b]

Missing value imputation

Spotted microarray data sets, in particular, often come perforated with missing values. Various spots may have been flagged as unreliable, for example due to scratches or debris on the slide, and therefore lacks expression values. In a study with many samples it is quite likely that a rather large fraction of the genes contain at least one missing value across samples. High-level analysis methods usually do not allow missing values so in order not to throw away too much potentially valuable information, the missing values need somehow to be filled in.

Different strategies to achieve this *missing value imputation* exists. A crude approach is to use the average expression value of the gene across samples. Another solution is adopted in *K nearest neighbor imputation* [Troyanskaya et al., 2001], where, if gene i contains a missing value in a particular sample, the K genes with most similar gene expressions in the rest of the samples (where the corresponding sample has a value) are found and the missing value is replaced by a weighted average of the values in the other genes.

Filtering

Filtering is often applied to a microarray data set prior to high-level analysis. By discarding genes that have noisy expression levels and/or do not vary significantly over samples it is believed that the performance of subsequent high-level analysis increases.

Different rules are applied in order to filter genes. For example, the Affymetrix^{TM} oligonucleotide microarray platform provides *detection p-values* estimating the confidence of the signal presence of each gene and filtering can thus be based on these p-values by requiring that a gene should be significantly present in at least a certain number of samples. For spotted microarray data, one can use similar criteria based on the ratio between foreground and background intensities or the fraction of missing values.

Furthermore, *variation filters* can be applied, excluding genes who, for example, have a ratio between standard deviation and mean value below some threshold value.

2.3.2 High-level analysis

Once data has been properly preprocessed, the next step is to extract some biological meaning from it. A multitude of tools from the fields of statistics, pattern recognition and machine learning are helpful for this purpose. This section reviews different types of methods that are adopted to extract different types of information.

At this point, it is useful to introduce a unifying conceptual framework and some notation within it. Generally speaking, the high-level analysis is a problem of mapping the expression data into some particular representation system, the choice of which will depend on the kind, and level, of structure we wish to infer.

First we need to settle how to mathematically represent the expression data set. Suppose that m measurements of the expression levels of n genes are given. Let x_{ij} be the estimated expression level of gene i in sample j and arrange the data in a matrix $\mathbf{X}$ where, thus, each row $\mathbf{g}_i$, $i = 1, \ldots, n$ represents the expression levels of a particular gene, and each column $\mathbf{x}_j$, $j = 1, \ldots, m$ represents the expression levels of a particular sample. We

may think of this set of data in two ways. The first is to look at the m samples as points in an n-dimensional *gene expression space* where the coordinates of a sample is given by the expression levels of its genes. Alternatively, we can consider the n genes as points in an m-dimensional space, the *sample expression space*, where the coordinates of a gene is given by its expression levels in the different samples. In this text, we will occasionally refer simply to the expression space and let the objects; genes or samples, indicate which space is intended.

While the mathematical representation of input data, as vectors in expression space is fairly obvious, the choice of representation of underlying patterns is more interesting. In clustering and classification, we commonly represent structure simply by class labels. An object (gene or sample) is thus described by a single integer, determining which partition of the objects it belongs to. Another structure representation is adopted in dimensionality reduction and regression, where data is mapped into Euclidean space, and thus each object is described by a set of coordinates in this space. The most complex structure representation we will discuss here is one commonly adopted in gene network inference, where objects (in this case, genes) correspond to nodes in some graph structure, and where the structure we wish to infer is the graph edges with their respective weights. To summarize, we may write:

$$\left.\begin{array}{cc} \text{gene expression space} & \mathbb{R}^n \\ \text{sample expression space} & \mathbb{R}^m \end{array}\right\} \ni x \longrightarrow z \in \left\{\begin{array}{cc} \mathbb{Z}_p & \text{clustering, classification} \\ \mathbb{R}^d & \text{dim. reduction, regression} \\ \langle V, E\rangle & \text{gene network inference} \end{array}\right. ,$$

where $\langle V, E\rangle$ are vertices and edges of a weighted graph.

In parallel to the framework described above, a common way to classify different problems in data analysis in general is to discriminate between *supervised* and *unsupervised* problems. Supervised problems assume the existence and use of prior knowledge, such as diagnosis, while unsupervised problems do not.

Microarray data sets have some significative features that have implications for what kind of information that can be extracted from it. First, they typically contain many more variables (genes) than observations (samples). Classical statistics typically assumes the study of a few variables, carefully chosen based on prior knowledge, to describe a particular phenomenon. For data from microarrays, as well as from several other emerging high-throughput measurement techniques, this is not the case. An abundance of variables is sampled, whereof, perhaps only a few, might be relevant. Second, due to the existence of genetic regulatory networks, there are complex dependence structures between genes, why the common assumption of independence between variables can not be made. These features lead to difficulties with using microarray data for stringent tests of hypotheses on a genome-wide level. However, testing hypotheses is not the only use one can have of data. *Generating* hypotheses is often equally valuable and it is in the light of this that gene expression data analysis should be viewed.

The rest of this section describes differential expression analysis, classification, clustering, dimensionality reduction and genetic network inference in gene expression data analysis.

Differential expression analysis

One of the most immediate questions in a study of a microarray data set is which genes are differentially expressed (DE) in two or more specified groups of samples. This problem is studied in *differential expression analysis*. Answers typically come in the form of gene lists which can be further studied in the search for biological insights to, for example, disease mechanisms.

For simplicity, suppose that we want to find DE genes in a two-group comparison. The most common approach is to study gene by gene and select those who show differential expression in the two groups. An early methodology for this is *fold analysis*, primarily designed for the case where only one sample per group is involved, as, for example, in a single spotted microarray hybridization with two competing query samples. The amount of differential expression is measured by the expression ratio between the two samples and differential expression is considered significant above and below constant threshold values, typically 2 and 0.5, respectively [Schena et al., 1995b].

Current data sets usually contain several samples per group. In this case, the use of statistical tests like some *t-test* becomes possible. For each gene, we may, for example, compute the two-sample t-statistic

$$t = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{s_1/m_1 + s_2/m_2}}$$

where $\bar{x}_1, \bar{x}_2$ are group means; s_1, s_2, estimated standard deviations and m_1, m_2, number of samples in group 1 and 2, respectively. The distribution of the statistic under the null hypothesis, which is that that the gene is not differentially expressed in the two groups, can either be assumed to follow a t-distribution or be estimated by permutation of the class labels. In this way, a p-value, quantifying the significance of the differential expression call, can be obtained.

Microarray data sets involve many more variables (genes) than observations (samples) and this needs to be taken into consideration while looking for DE genes. For example, consider a data set where the number of genes is $n = 10^4$ and the fraction of truly non-DE genes is $\pi_0 = 0.9$. Suppose $p = 0.05$ is chosen as a threshold for calling a gene differentially expressed. The expected number of non-DE genes that is incorrectly called differentially expressed is then $pn\pi_0 = 450$ — almost half the number of truly DE genes. Moreover, not all DE genes will be called DE since, in some cases, by chance, the p-value will exceed 0.05. The result is that the gene list will contain a large fraction of genes that have nothing to do with the condition that defines the groups. This exemplifies the *multiple testing problem*, where a p-value threshold that seems standard and reasonable in a single test leads to a high *false discovery rate (FDR)*, defined as the expected proportion of false positives among the declared significant results [Benjamini and Hochberg, 1995]. A better way of selecting the DE genes is therefore by controlling the FDR instead of the p-value [Storey and Tibshirani, 2003, Pawitan et al., 2005]. Related to this problem is also the estimation of the true proportion of DE genes [Broberg, 2005].

The t-test is not always the optimal choice of test for differential expression. If expression distributions are non-normal, for example in the presence of outliers, other tests, like the Wilcoxon test, is more suitable. Another problem is that a small number of samples makes it difficult to get significant results in the tests. A third problem is that gene ex-

pressions are typically not independent, so the assumption of independence and identical distribution is violated. However, if genes are, on average, uncorrelated and the dependencies are weak, it can be argued that this problem is less severe[Storey and Tibshirani, 2003, Pawitan et al., 2005].

Treating genes one-by-one does not make use of the full potential in a microarray data set. Since groups of genes are correlated it makes sense to perform differential expression analysis of whole groups of genes instead of single genes. Such groups may, for example, be defined as genes known to be involved in particular pathways of interest or as genes located in the same chromosomal regions. *Gene set enrichment analysis* [Subramanian et al., 2005] implements a method for this, and it has been shown that, using this method, differential expression of groups of genes can be identified where single gene tests would fail to discover differential expression.

Classification

The task of *classification* is that of learning how to best guess which, out of a number of given classes, an object with unknown class label belongs to. A classifier is constructed by training, where it is introduced to representative training objects of known classes — the *training set*. For microarray data, the applications of classification include diagnosing cancer type, given the expression pattern from a tumor sample, or predicting the biological function of genes based on their expression patterns.

In a sense, classification is similar to differential expression analysis, since most classification algorithms, explicitly or implicitly, works by finding variables, or functions of variables, that are good predictors of class. A difference, however, is that while a biological interpretation of these predictors is a nice side-effect, it is not the primary goal as in differential expression analysis. The similarity is particularly clear in a class of methods represented by Golub et al. [1999], where, given a two-group classification problem, a list of differentially expressed genes is extracted using the training set, and a voting function, defined on these genes, decides which class a presented sample belongs to. A problem with this approach is that many of the discriminative genes are likely to be correlated, perhaps being involved in the same particular process. Genes with less strong differential expression, but uncorrelated with the group of most strongly DE genes, would most likely increase the generalization performance but are not included using the basic method described.

In gene list based classifiers, as above, the decision function is fixed and predefined while the set of variables on which it operates is learned from the data. Other methods, such as *Artificial Neural Networks (ANNs)* and *Support Vector Machines (SVMs)*, use all variables but lets the method learn the decision function from the data. Here too, a list of predictive variables can be extracted by ranking them according to their influence on the final decision function. Artificial neural networks can readily be applied to classification problems with more than two classes [Khan et al., 2001], while support vector machines are binary classifiers, discriminating between, for example, healthy and cancerous tissue [Furey et al., 2000], or classifying genes as belonging to a known functional group [Brown et al., 2000] or not. Binary classifiers can be extended to handle K classes by learning to classify between each pair of classes or by learning K classifiers of one class against all others. Both ANNs and SVMs are able to learn nonlinear decision functions. In SVMs

this is achieved by a *kernel transformation* of the data, something that will be discussed in more detail in Section 3.4.

Just like in differential expression analysis, the fact that the genes by far outnumber the samples, introduces some difficulties. For example, gene list based classifiers, as described above and in [Golub et al., 1999], classify judging from lists of top discriminatory genes. As pointed out in the discussion on differential expression analysis, such lists are likely to be hampered by false positives, which by their presence disturbs the classification of new samples. In fact, this problem is common to all classification methods when the number of samples is much smaller than the number of variables — A classifier risks to learn to trust variables or sets of variables as having predictive power when in fact it only seems so by chance.

A wealth of different classification methods have been applied to microarray data but, so far, none has stood out as significantly more suitable than the others. Presumably, the importance of the method choice will grow with the number of samples in the data sets [Dudoit and Fridlyand, 2003].

Clustering

Clustering is the process of grouping together similar objects into resulting groups, or clusters. In gene expression data analysis, clustering serves to discover groups of co-regulated genes or groups of samples with similar expression profiles, for example revealing classes or subclasses of disease states. The problem is similar to that of classification, with the difference that while clustering methods discovers groups in data without using any prior knowledge, classification methods does so when arranging objects into groups. Indeed, classification methods are sometimes referred to as *supervised clustering*.

The problem of grouping together 'similar' objects calls for a definition of similarity. In Section 2.4, this subject is discussed more thoroughly, while for now we note that two common ways of defining (dis)similarity is by means of Euclidean distance and correlation distance, defined as $1 - \rho_{xy}$, where ρ_{xy} is the correlation between vectors x and y.

Hierarchical clustering. *Hierarchical clustering* is currently the most frequently used clustering method in gene expression data analysis (Eisen et al. [1998] gave an early example). The (agglomerative) hierarchical clustering algorithm takes as input a matrix of pairwise similarities between objects. Initially all objects are considered as clusters. Then, iteratively, the most similar cluster pair is found and merged together into a new cluster. This is repeated until all objects are contained in a single cluster. Similarities between two clusters can be defined in a number of ways, for example, the largest similarity between any pair of objects in separate clusters (*single linkage*), the smallest similarity between any pair of objects in separate clusters (*complete linkage*) or the average similarity between all pairs of objects in separate clusters (*average linkage*). The clustering is visualized in a cluster tree, a *dendrogram*, visualizing the nestled structure of clusters. Hence, in fact, hierarchical clustering yields a more detailed structure representation (a tree-graph) than many other clustering methods that simply divide data into partitions (cf. Section 2.3.2).

Hierarchical clustering is frequently used in comparative genomics and phylogeny to study, for example, the evolutionary development of gene sequences, and perhaps hierarchical clustering is more suited for data where distances can be defined as a discrete number

of alterations, than for quantitative data like gene expression data. One problem is that it might be difficult to decide which clustering level in the dendrogram to choose, if the aim actually is to partition the data. On the other hand, the user does not have too provide an a priori number of clusters. Another issue is one of over-fitting. Different ways of defining the similarities between points and clusters yield very different cluster trees. Hence, there is a risk of adjusting parameters until getting a tree that adhere to prior beliefs.

K-means clustering. *K-means clustering* [Forgy, 1965] is a standard and well understood clustering algorithm. The algorithm takes as input the expression data and the number of clusters, K. Initially, K cluster centers are randomly placed in the span of the data. All objects are assigned to their nearest cluster center and the mean expression of each cluster is calculated. These means replace the prior cluster centers and the two steps are repeated until convergence.

The advantages with K-means is that the method does not have many parameters to assign, while in many cases it is a drawback that the number of clusters has to be provided to the algorithm. An example of the use of K-means clustering in gene expression data analysis can be found in [Tavazoie et al., 1999].

Self-organizing maps. The *self-organizing maps (SOM)* algorithm [Kohonen, 1982] is algorithmically reminiscent of K-means clustering. A significant difference lies in the way results are presented. The resulting clusters are ordered in a, usually two-dimensional, grid, a *feature map* so that neighboring clusters in the feature map are more similar than clusters far from each other. Further, statistical properties of the data are reflected in the feature map. Regions in data space with a high density of points are mapped onto larger domains in the feature map. Because of this ability to display continuous features of the data space, SOMs can be seen as a hybrid between clustering methods and dimensionality reduction methods, discussed in the next section. A drawback of the method is that several parameters need to be specified. Tamayo et al. [1999] demonstrated the use of self-organizing maps in gene expression data analysis.

Important to keep in mind is that most clustering algorithms will divide data into clusters even if no real cluster structure is present. Therefore, it is important to control the significance of the produced results. For example, when clustering genes, measured over a small number of samples, the clusters will contain many false positives, while many true positives will be missed. This is due to the fact that the statistical confidence of similarity measures will be low.

Dimensionality reduction

Clustering methods divide objects into discrete groups. Sometimes such a description of the data is less relevant, in particular if the properties of the objects vary continuously with respect to some underlying parameter. *Dimensionality reduction* methods provide means of representing objects in low dimensions with an aim of revealing such parameters. Generally, dimensionality reduction methods learns some function from objects described by many variables to representative objects described by few variables, in such a way that important properties of the data are optimally conserved. With the possibility to display

results in a transparent way, as two- or three-dimensional scatter plots, dimensionality reduction methods are well suited for explorative data analysis.

Dimensionality reduction is typically used for visualization of patterns in data. In gene expression data analysis, two- or three-dimensional visualizations may be inspected in order to discover outliers or to give rise to hypotheses about groups, or subgroups, of samples or genes. The links between clinical variables and observed patterns can be investigated. Moreover, if patterns can be inferred to be related to non-biological parameters such as laser settings or hybridization date, data has likely not been sufficiently well normalized. Hence, dimensionality reduction can also be used as a means of data quality control. Another use of dimensionality reduction is as a compressive preprocessing prior to clustering or classification, with the intentions to filter out noise or to relief the computational burden of subsequent methods by reducing the number of input variables.

Standard methods of dimensionality reduction include *principal component analysis* and *multidimensional scaling*, and these are also the ones primarily adopted within gene expression analysis [Alter et al., 2000, Khan et al., 1998, Hedenfalk et al., 2001, Andersson et al., 2005b]. These methods work best when the underlying data patterns are linear, and are not designed for data where the dependencies between variables are nonlinear. This work aims to study various aspects of the application of nonlinear dimensionality reduction methods, as reviewed in Chapter 3, to gene expression data.

Gene network inference

As discussed in Section 2.1, genes interact with each other in regulatory networks. It is therefore natural to argue that the most biologically authentic representation of the genes is, not as a number of clusters, neither as a point cloud in Euclidean space, but instead as a network describing the functional relations between genes.

There are many possible ways to model the interactions between genes in the regulatory networks [de Jong, 2002], ranging in complexity from simple *Boolean Networks* [Kauffman, 1969, 1993], modelling regulatory networks as random directed graphs, to detailed *Stochastic Master Equation models* [Arkin et al., 1998], modelling the dynamics of the probability distributions of all the individual molecule species abundances. For genetic network inference, the model has to be chosen with respect to the relation between the size of the system and the available number of measurements. If we are interested in studying the whole genome, inference of detailed models is unfeasible, and one has to resort to simpler representations.

Here we will concentrate on linear ODE models, for which inference of model parameters for whole-genome systems is computationally feasible. To this end, the expression level $g_i(t)$ of gene i is typically modelled by

$$\frac{dg_i}{dt} = \sum_k w_{ik} g_k(t) + b_i(t) + \varepsilon_i(t), \tag{2.2}$$

where $\varepsilon_i(t)$ are noise terms; $b_i(t)$, external stimuli and w_{ik}, weights that determine the influence of gene k on gene i.[4] Naturally, it is not likely that such a model can capture

[4] Note that degradation of the gene product i can be taken into account in the term w_{ii}.

all aspects of the network dynamics, but under the assumption that the system is close to steady-state, (2.2) can be taken as the linearization of some more general nonlinear model.

The aim of gene network inference under the model (2.2) is thus to estimate w_{ik} given m measurements of the n gene expressions levels. Since generally $n >> m$, this becomes an under-determined problem, meaning that multiple solutions will exist. In other words, finding a solution is generally not difficult — the challenge is finding a biologically relevant one. Different approaches have been taken to this problem. For example, van Someren et al. [2000] cluster the genes into m clusters and infer a model for the dynamical interaction between the clusters, a problem that is more likely to have a unique solution. A related solution is proposed in [Holter et al., 2001], where dimensionality reduction through principal component analysis is applied and weights for a system consisting of the m most variant modes instead of the n genes are inferred. Other approaches do not attempt to reduce the number of interacting entities, but instead make use of some, biologically motivated, additional constraint to infer a realistic model. For example, in [Yeung et al., 2002] the set of all valid solutions is computed through singular value decomposition and then an optimally sparse solution matrix is found using L_1 regression. The sparseness criterion is biologically motivated by the well accepted assumption that genetic regulatory networks are sparse, that is, that each gene only interacts with a few others. Also adopting the sparseness criterion, Gustafsson et al. [2005] use Lasso regression [Tibshirani, 1996] to infer the genetic network.

2.3.3 Further up the chain of analysis

Once high-level analysis methods have suggested some underlying structure in the data, these results need to be interpreted and validated in terms of biological significance. This can be done in a number of different ways. Suppose, for example, that we have clustered the data, so that what we have to validate is a particular partition of the data.

A natural way to validate sample clusters is to consult clinical variables of the samples (e.g., gender, blood pressure, cancer diagnosis) and investigate if patterns, similar to the ones discovered in the gene expression data, appear there. With an extensive clinical database this might be a task on almost the same complexity level as the gene expression analysis itself. One particular way of evaluating proposed sample clusters in diseases like cancer is the use of *Kaplan-Meier survival analysis*, which involves a statistical test of whether two groups of patients have significantly different median survival times. If this is found, it is often argued that the clusters are of biological relevance, for example, as subgroups of the same disease [Alizadeh et al., 2000].

For genes, the available knowledge does not come in the shape of sets of clinical variables like for samples. Instead, gene clusters can be validated with respect to, e.g., cellular functions, chromosomal locations, sequence information, etc. Knowledge about the genome is stored in *Gene Ontology (GO)* [Ashburner et al., 2000] databases where genes are arranged in tree structures according to function, location and other properties. Given a set of genes, one can make a query to a GO database testing if some, say functional, group on some tree level is over-represented among the genes [Zhong et al., 2003]. Alternatively, databases containing known pathway relations, such as the *Kyoto Encyclopedia of Genes and Genomes (KEGG)* [Kanehisa et al., 2006], may be consulted to see whether genes from some particular pathway are over-represented in the cluster. The evaluation of the mean-

ing and relevance of gene clusters using queries to databases requires that enough useful information has been stored in the database by human curators. To sidestep this, one may make use of *text mining* techniques which search through vast collections of literature and attempt to extract relevant information. For a given gene cluster, text mining methods can retrieve abstracts where subsets of the given genes co-occur. From these abstracts, over-represented keywords can be extracted in order to gain understanding of the functional or clinical context of the genes [Blaschke et al., 2001]. On a more detailed level, the literature can be mined for causal relations between the genes, thereby suggesting a network of functional relations between genes [Jenssen et al., 2001]. A third way to evaluate gene clusters is to consult sequence data. The upstream regions of the genes are then searched for shared motifs, presumably corresponding to known or unknown transcription factors [Roth et al., 1998]. The existence of such shared motifs may then confirm the biological relevance and aid the understanding of the role of the gene cluster.

Even if the discussion above assumes that the inferred structure is in the form of clusters, the same sources can, in principle, be used to evaluate continuous or graph representations of structure, given the appropriate modifications.

2.4 Gene expression space

In this section we discuss some properties of gene expression data within the expression space framework presented in Section 2.3.2. We also state the central assumption that this work is based on.

Recall that we represent a set of m microarray samples measuring the expression level of n genes, either as m samples $\mathbf{x}_j$, $j = 1, \ldots, m$ in n-dimensional gene expression space, or as n genes $\mathbf{g}_i$, $i = 1, \ldots, n$ in m-dimensional sample expression space.

Assuming that microarray measurements give quantitatively relevant information about the underlying biology is the same as assuming that the location of a gene in sample expression space or a sample in gene expression space is related to its biological properties. In gene expression space, we therefore interpret different domains as representing different cellular states, linked to biological phenotype, such as tumor state, cell cycle phase, etc. Similarly, the time development of a cellular system is a trajectory through gene expression space.

The interpretation of the sample expression space is a bit less obvious. Here different regions correspond to genes that have similar behavior over the samples in the data set, so that, for instance, in time-series experiments, they co-vary in time, and in cross-sectional studies they behave similarly in all conditions. The specific interpretation of a given sample expression space therefore depends on the selection of cell samples that were included in the study. While discussing sample expression space, it is worth mentioning that the standard normalization assumption that a majority of genes remain unchanged over samples is equivalent to the assumption that the majority of genes lie along the main diagonal in expression space, or, if data was preprocessed by mean-centering the genes, close to the origin.

In the discussion above, the meaning of similarity in expression space has not been closely specified. Most commonly, (dis)similarity is measured in terms of Euclidean dis-

tance, which, for two vectors $\mathbf{x}, \mathbf{y}$ is

$$d_{Euc}(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{k=1}^{n} (\mathbf{x}_k - \mathbf{y}_k)^2},$$

or in terms of correlation distance

$$d_{Corr}(\mathbf{x}, \mathbf{y}) = 1 - \frac{\sum_{k=1}^{n} (\mathbf{x}_k - \bar{\mathbf{x}})(\mathbf{y}_k - \bar{\mathbf{y}})}{\sqrt{\sum_{k=1}^{n} (\mathbf{x}_k - \bar{\mathbf{x}})^2 \sum_{k=1}^{n} (\mathbf{y}_k - \bar{\mathbf{y}})^2}},$$

where $\bar{\mathbf{x}}$ denotes the mean of $\mathbf{x}$.[5] Euclidean distance measures the straight line distance between points, while correlation distance measures (1 minus the cosine of) the angle between points. For applications in gene expression analysis, the distance metric should be chosen so that it is optimally biologically relevant with respect to the study at hand. In other words, small distance should correspond to high biological similarity and large distance to low biological similarity. Different distance measures capture different aspects of similarity, and each have their particular strengths and limitations. As an example, consider a pair of genes with small magnitudes, but with qualitatively different expression patterns over samples. These will lie close to the origin but at different angle, so in Euclidean distance they would be close while in correlation distance they would be far apart. On the other hand, consider two genes with expression patterns of [10000000] and [00000001] where the ones might be outlier values. In this case, perhaps the Euclidean distance is more reasonable with its judgement that the genes are close than the correlation distance which says that the genes are far.

The dependencies between genes through genetic regulatory networks, as discussed in Section 2.1, imposes functional relations between them which restricts[6] the possible cellular states to some manifold, or hyper-surface, in gene expression space.

If expression data lies on a Riemannian manifold, a natural and biologically relevant choice of distance metric would be the geodesic distance on the underlying manifold, that is, the distance travelled by an insect, taking the shortest path between two points on the manifold. The geodesic distance is similar to the Euclidean on a small scale but may differ widely for larger distances since the manifold may be curved. Another way of stating the problem with Euclidean distance in this setting is the following observation: two points of the set are either close, and then their Euclidean distance reflects useful information, or they are far from each other and the meaning of their Euclidean distance is different in different directions for the reason that some domains in expression space cannot be crossed by the real system. This is summarized in the following central assumption which motivates our work:

> Data are sampled from, or reasonably close to, a Riemannian manifold in expression space whose geodesic distance metric is biologically relevant.

Having stated this, we need to recognize some complications with this view concerning what is actually observed. On a practical level, there are difficulties due to noisy data, high dimensionality, and the non-trivial relation between real expression levels and their

[5] If the vectors are mean centered and scaled to unit variance, the two distances are related by $d_{Euc}^2 = 2 d_{Corr}$.

[6] disregarding noise

microarray estimates. There are also some more fundamental problems. Despite their ability to measure abundances of huge sets of mRNA molecules, microarrays do not let us study all relevant state variables of the genetic regulatory system. Other types of molecules need to be observed, the most important being the proteins, in order to get a complete observation of the whole system.

Nevertheless, microarray studies have shown that a lot of biology is reflected in the subset of state variables that the mRNA molecules constitute, why there are good reasons to believe that our central assumption above is relevant and useful. Thus, if the metric on the manifold can somehow be estimated from the data, valuable pieces of information are attained. Estimated geodesic distances can be used for clustering and classification purposes, as well as for dimensionality reduction which is the focus of this thesis. In a sense, dimensionality reduction ideally allows us to get a glimpse of the manifold in gene expression space, and to visualize some of its underlying parameters.

The next chapter gives a general review of dimensionality reduction. Chapter 4 presents results demonstrating that nonlinear dimensionality reduction based on approximate geodesic distances yields more biologically relevant representations than linear methods. Similar results are also shown in Chapter 5 where an alternative way of computing approximate geodesic distances is applied. Chapter 6 discusses what these results imply concerning the validity of the assumption above that gene expression data are samples from a Riemannian manifold in expression space.

Chapter 3

Dimensionality reduction

3.1 Introduction

As our technological means to measure and store information of our environment improve, we face the need to manage more and more objects described by more and more variables. The previous chapter describes just one out of many such cases.

A range of techniques exist that provide measurements, on varying spatial scales, of the neurological activity in the brain. The neural system activity is sampled in time and space, allowing us to peak into one of the most intriguing systems in nature. Examples of such techniques include Electroencephalography (EEG), Functional Magnetic Resonance Imaging (fMRI) and intracellular electrophysiological recording techniques.

Grayscale images of the size $M \times N$ pixels may be seen as vectors in an $M \cdot N$-dimensional pixel space and RGB color images, having three intensity values at each pixel, similarly as $3 \cdot M \cdot N$-dimensional vectors. Taking this to the extreme, multi- and hyperspectral imaging collects spectral information across, not only three, but thousands of spectral bands, thus increasing the image data dimensionality further. Alternatively, instead of studying images as objects, we can study the individual pixels which are now described by P-dimensional vectors, where P is the number of spectral bands. Multi- and hyperspectral imaging can, for example, be used to register information about chemical composition of such different things as melons [Nakauchi, 2005] and planetary nebulae [Mekarnia et al., 2004]. Films may also be seen as multidimensional data. Given T frames of an $M \times N$ pixel grayscale film we can represent it as a point in $T \cdot M \cdot N$-dimensional space.

Text documents, such as web sites or scientific articles, may be described by individual frequency counts of each word in a thesaurus of tens of thousands of words. In this way, we may view the world wide web as a gigantic point cloud of documents in word space.

Financial time-series of stock and currency rates as well as transaction data have representations as high-dimensional vectors. Another example is consumer data bases, where personal information is gathered together with purchase records.

The focus of this thesis is on the analysis of microarray data, but biotechnology offers many other massive data sources. For example, various proteomics technologies such as mass spectroscopy, electrophoresis gels and antibody chips produce data on the individual

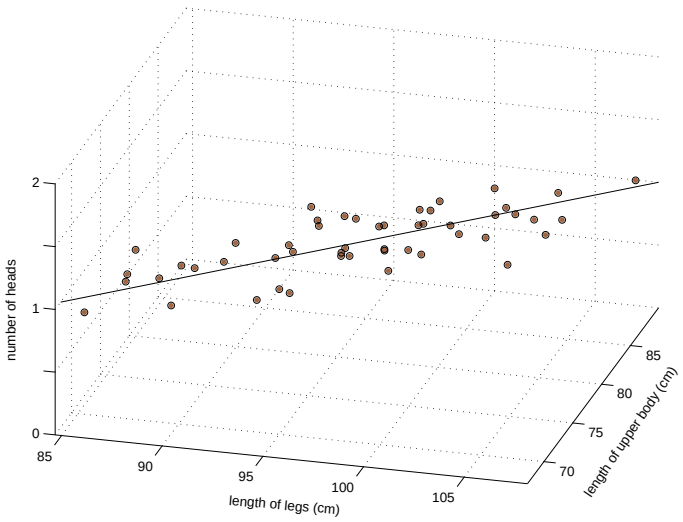

Figure 3.1: Anatomical measurements of 50 subjects. The one-dimensional orthogonal projection plane that captures most variance is drawn as a line.

abundances across a wide range of proteins in given cell samples.

The fact that we can collect information about the world with increasing resolution is of course positive, but alone it does not aid much to our understanding. Underlying patterns of importance often involve a complex interplay between several variables and, furthermore, many of the variables do not carry any relevant information. In other words, more potentially useful information is gathered but it is intricately hidden in a mass of less useful information. To this end, we need computational methods to extract underlying relevant patterns and to obtain an overview of the data, possible to grasp for low-dimensional creatures like us. *Dimensionality reduction* does this. Point configurations in low-dimensional space are constructed so that their coordinates reflect some given aspect of the high-dimensional data.

The dimensionality reduction problem may be posed as one of learning functions defined on the original data, taking it into lower dimension and filtering out relevant features. One example in this respect is *projection pursuit* methods who apply orthogonal projections to data and search for the projection that maximizes some *projection index* quantifying how interesting a projection is. Examples of projection indexes include *Fisher information*, *negative Shannon entropy* and the *univariate Friedman-Tukey index* (see [Carreira-Perpinan, 1997] and references therein). The, without comparison, most popular projection pursuit method is *principal components analysis (PCA)*, where the projection index is the variance of the resulting projection. The usefulness of variance as a projection index is illustrated by the following example:

Example 1. *Imagine that you have just witnessed a bank robbery. The police have arrived at the scene and you, as a witness, are asked to describe the perpetrator. Now, you have the choice*

between many possible variables that might describe the bank robber. How do you choose? The police officer will not likely be very satisfied with the answer "He had one head." or "He had two legs.". Obviously this holds true for the vast majority of possible bank robbers. Instead the officer would be more happy if you provide him with the values of variables in which the group of possible perpetrators show a larger variance such as height, hair color, eye color, etc.

Apart from showing why variance is important, the example above illustrates a problem of *variable selection*, where no regard is taken to possible dependencies between variables. The fact that PCA searches for orthogonal projections of variables with maximum variance means that it detects specific linear combinations of variables along which the data varies maximally.

Example 2. *Suppose that we, in the search for variables that lets us discriminate between people, such as bank robbers, measure the three variables [number of heads], [leg length] and [upper body length] for a number of test subjects. A scatter plot of the data is shown in Figure 3.1 together with the one-dimensional orthogonal projection plane (line) that captures the most variance. This line catches a large fraction of the total variance why we may conclude that there is only one underlying variable of importance. Furthermore, the equation of the line tells us that this underlying variable is approximately a sum of [leg length] and [upper body length] with [number of heads] being constantly 1. Of course, the underlying variable is the total body length.*

Principal components analysis has favorable computational properties compared with many other projection pursuit methods since, as will be shown in Section 3.3, it has an analytical solution and does not require difficult optimization schemes that risk finding local minima. Because of its simplicity and utility it has become widely spread and known across numerous disciplines.

As demonstrated above, variance often reflects interesting patterns in the data. One should be aware, however, that it does not provide a universal measure with which we can find all interesting patterns.

Example 3. *Consider the data in Figure 3.2. Here the largest variance is along the y-axis but perhaps the x-axis says more about the data since here it is separated into two groups.*

Even if variance is indeed an appropriate projection index for the data and the problem at hand, there are many cases where PCA is insufficient for the reason that it restricts the class of dimensionality reducing functions to orthogonal projections. The underlying functional relation between variables is not always linear. Under such circumstances, it is not sufficient to find the optimal orthogonal projection of data, as illustrated by the following example.

Example 4. *Assume that we are measuring properties of a mechanical system where a body of mass m experiences a gravitational force f at a distance r from a body with fixed unit mass. Then our data set consists of samples $x_i = [m_i, r_i, f_i]$. We know from classical mechanics that*

$$f \sim \frac{mM}{r^2},$$

where M is the mass of the fixed body, in this case 1. Our samples x_i thus lie on a curved two-dimensional surface in $\mathbb{R}^3$ (see Figure 3.3). Since the variance is along a curved surface, no

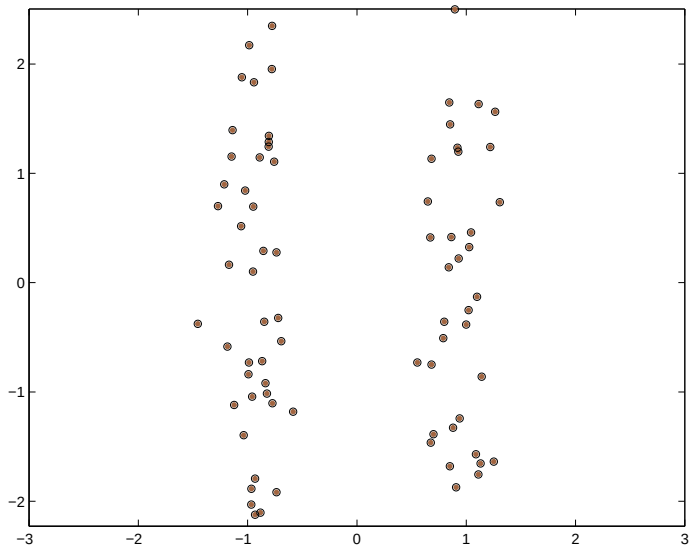

Figure 3.2: A data set for which variance is not an optimal projection index.

orthogonal projection from $\mathbb{R}^3$ *to* $\mathbb{R}^2$ *can fully explain all the variance. Ideally, the projections should instead be onto the surface* $f = m/r^2$.

Several approaches have been taken to extend PCA and other linear dimensionality reduction methods to handle nonlinear data. An intuitively natural generalization of PCA is *Principal Curves* [Hastie and Stuetzle, 1989], where curves, instead of straight lines, through the data is found. A principal curve should be such that the sum of squared distances from the points to the curve is minimized under some given smoothness constraint. If maximum smoothness is required, the curve becomes a straight line given by the PCA solution. A selection of other methods for nonlinear dimensionality reduction includes *Principal Manifolds* [Smola et al., 1999], *Generative Topographic Mapping* [Bishop et al., 1998], *Manifold Charting* [Brand, 2003], *Stochastic Neighbor Embedding* [Hinton and Roweis, 2003] and *Nonlinear Auto-encoders* [Diamantaras and Kung, 1996].

In this chapter we will review a number of approaches that may be classified as *spectral dimensionality reduction methods*. These methods have in common that they rely on the spectral decomposition of some matrix estimating some geometrical property of the data[1], thereby providing relatively efficient ways to obtain globally optimal solutions compared to methods based on iterative optimization.

The following section defines the theoretical framework within which we put the dimensionality reduction methods. Then PCA is described followed by a derivation of its formulation as a *kernel method* yielding *kernel PCA* [Schölkopf et al., 1998]. Chronologically preceding kernel PCA, but with many similarities is the extensive field of *multidi-*

[1] For completeness of the survey, we will sometimes go astray from spectral methods, for example in the case of non-classical MDS in Section 3.5.

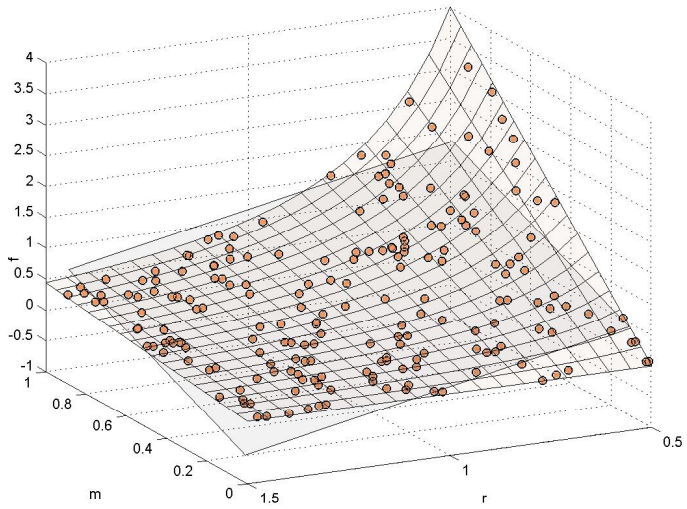

Figure 3.3: Different measurements on a two-body mechanical system. The nonlinear variable dependence leads a curved data distribution, shown by the wire-frame. The two-dimensional PCA projection plane, shown by the transparent plane, is unable to fully represent the underlying two-dimensional pattern.

mensional scaling (MDS), which is described in Section 3.5. Rather than working with coordinate data as standard PCA does, multidimensional scaling works takes dissimilarity data as input. Following MDS we will cover some more recent approaches which all have in common that they set out to approximate the underlying manifold by an adjacency graph, connecting neighboring data points. In other words, the manifold — a continuous object, is approximated by a discrete object — the adjacency graph. Generally, the aim is then to make use of geometrical properties of the manifold for which graph counterparts can be readily computed and summarized in a matrix whose spectral decomposition yields a lower-dimensional embedding. For example, the *Isomap algorithm* [Tenenbaum et al., 2000], presented in Section 3.6, approximates geodesics on the manifold by graph geodesics while *Laplacian Eigenmaps* [Belkin and Niyogi, 2003], in Section 3.7, approximates the Laplace-Beltrami operator on the manifold by the graph Laplacian. *Locally Linear Embedding* [Roweis and Saul, 2000], described in Section 3.8, estimates linear representations of local neighborhoods while *Hessian Eigenmaps* [Donoho and Grimes, 2003] computes approximations of the Hessian operator. We will refer to these methods as *graph-based methods* for spectral dimensionality reduction. Many of them can be formulated as kernel PCA with kernels learnt from data and we will conclude the chapter in Section 3.9 with a discussion on this.

3.2 Framework

Suppose that we are given a set of vectors $\hat{\mathcal{X}}_m = \{x_1, x_2, \ldots, x_m\} \subset \mathbb{R}^n$. Furthermore, assume that these vectors are samples from a manifold $\mathcal{X} \subset \mathbb{R}^n$. We will refer to $\hat{\mathcal{X}}_m$ as the *input coordinates*, to $\mathcal{X}$ as the *observation manifold* and to $\mathbb{R}^n \supset \mathcal{X}$ as the *input space*. A relevant question is what can be said about the observation manifold given only the input coordinates. To specify this problem further we need to decide which properties of $\mathcal{X}$ that are relevant. For example, we could be interested in the topology of the manifold. However, for our purpose, the topology does probably not contain as much information as we would like to extract. If, for example $\mathcal{X}$ was a two-dimensional manifold, the problem would boil down to determining the genus of $\mathcal{X}$, and no difference would be made between homeomorphic manifolds having widely different input coordinates. Instead it is more useful to seek the metric of the manifold, since this provides us with a way to quantify the similarity between points on $\mathcal{X}$, similarities that presumably carry relevant information about the physical object that is studied. To summarize, so far, our main assumption is that

> the input coordinates $\hat{\mathcal{X}}_m$ lie on a Riemannian manifold $\mathcal{X}$ whose metric tensor g carries relevant information.

The problem of *manifold learning* is to

> given input coordinates $\hat{\mathcal{X}}_m \subset \mathcal{X}$, determine $(\mathcal{X}, g)$, where g is the metric tensor.

As stated, this is, in fact, impossible given a finite number of input coordinates. To see this, note that we may always fit a curve to $\hat{\mathcal{X}}_m$ passing through all points, so additional assumptions about $\mathcal{X}$, such as its intrinsic dimension, need to be made, in order to make the problem well-posed.

Related, but not identical to manifold learning is our definition of the *dimensionality reduction* problem:

> Find a mapping $\Psi : \mathcal{X} \to \mathcal{Z} \subset \mathbb{R}^p$, where $\mathcal{Z}$ is an affine space of dimension $p < n$, such that $\hat{\mathcal{Z}}_m = \{z_1, \ldots, z_m\} = \Psi(\hat{\mathcal{X}}_m)$, in some sense, represents the metric structure in $\hat{\mathcal{X}}_m$ well.

The point configuration $\hat{\mathcal{Z}}_m$ is denoted the *reconstructed embedding coordinates*, or in short a *reconstruction* of $\hat{\mathcal{X}}_m$. Similarly, we call the space $\mathcal{Z}$ the *reconstructed embedding space*, or alternatively, the *feature space*. The mapping Ψ is, in some methods, learnt explicitly, but in most cases it is found implicitly, and point-wise, by computing $\hat{\mathcal{Z}}_m$ directly. Dimensionality reduction is not the only problem that can be formulated as one of learning functions defined on the input coordinates. The problems of classification and regression can also be formulated in a corresponding way.

A common assumption is that $\mathcal{X}$ is the image of an affine Euclidean space $\mathcal{Y}$ under a smooth bijective mapping Φ, possessing some given properties. In these cases, we refer to $\mathcal{Y}$ as the *parameter space*, and $\hat{\mathcal{Y}}_m = \Phi^{-1}(\hat{\mathcal{X}}_m)$ as the *embedding coordinates*.

The term *nonlinear dimensionality reduction* is used to stress the fact that Ψ is not restricted to be a linear mapping (cf. Example 4). Finally, dimensionality reduction methods

may be divided into *local methods*, attempting to create reconstructions that are locally coherent with the structure on $\mathcal{X}$ and *global methods*, aiming at creating reconstructions that are correct on all scales.

3.3 Principal component analysis

Principal Component Analysis (PCA) is, by far, the most widely used method of dimensionality reduction. The method relies on an eigendecomposition of the covariance matrix. Historically, PCA goes back to Pearson [1901] and was further popularized by Hotelling [1933]. Thorough accounts of the theory and methodology can be found in [Jolliffe, 1986]. Since many of the other dimensionality reduction methods that will be presented later in the chapter can be seen as different variants and generalizations of PCA we will, in this section, go into some detail while describing the method.

3.3.1 Finding optimally variance preserving projections

Let $\mathbf{X} = [\mathbf{x}_1 \dots \mathbf{x}_m]$ be the *data matrix* having the observed input coordinates, scattered in $\mathbb{R}^n$, as column elements and define the *covariance matrix* of the variables as

$$\mathbf{C}_{jk}(\mathbf{X}) = \sum_{i=1}^{m} (\mathbf{X}_{ji} - \frac{1}{m}\sum_{i=1}^{m} \mathbf{X}_{ji})(\mathbf{X}_{ki} - \frac{1}{m}\sum_{i=1}^{m} \mathbf{X}_{ki}), \qquad j,k = 1,2,\dots,n.$$

Without loss of generality, we may assume that data is centered, i.e., $\sum_{i=1}^{m} \mathbf{x}_i = 0$, and thus write $\mathbf{C} = \mathbf{X}\mathbf{X}^T$. The $n \times n$ matrix $\mathbf{C}$ is positive and symmetric so by the spectral theorem there exist nonnegative eigenvalues $\mu_1 \geq \mu_2 \geq \dots \mu_n \geq 0$ and a corresponding orthonormal base of eigenvectors $\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_n$ in $\mathbb{R}^n$ such that

$$\mathbf{C}\mathbf{s}_k = \mu_k \mathbf{s}_k, \qquad k = 1,2,\dots,n.$$

In the rest of this section we will assume that $m < n$, and thus $\mu_{m+1} = \dots = \mu_n = 0$. In this case, $\mathbf{s}_{m+1}, \dots, \mathbf{s}_n$ may be found by, for example, Gram-Schmidt orthogonalization. Let $\mathbf{S}$ denote the $n \times n$ matrix having $\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_n$ as columns, and let λ_k, $k = 1,2,\dots,n$ be the singular values of $\mathbf{X}$, i.e., $\lambda_k := \sqrt{\mu_k}$.

The *principal components* of the data set are then the set of unit length vectors in $\mathbb{R}^m$ defined as

$$\mathbf{p}_k := \frac{1}{|\mathbf{X}^T \mathbf{s}_k|}\mathbf{X}^T \mathbf{s}_k = \frac{1}{\lambda_k}\mathbf{X}^T \mathbf{s}_k \qquad \text{for all } k = 1,2,\dots,n \text{ such that } \lambda_k \neq 0.$$

Lemma 1. *The principal components constitute an orthonormal set.*

Proof. Since $\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_m$ is an orthonormal set it follows that

$$\mathbf{p}_i{}^T \mathbf{p}_j = \frac{1}{\lambda_i \lambda_j}\mathbf{s}_i{}^T \mathbf{X}\mathbf{X}^T \mathbf{s}_j = \frac{\mu_j}{\lambda_i \lambda_j}\mathbf{s}_i{}^T \mathbf{s}_j = \delta_{ij},$$

for i, j fixed. □

Define an orthogonal $m \times m$ matrix P having the principal components $\mathsf{p_k}$ as column elements and an $n \times n$ diagonal matrix Λ having, as diagonal elements, the singular values of X. Moreover, denote by Λ_m the upper left $m \times m$ submatrix of Λ. From the definition of the principal components and Lemma 1 it follows

$$\mathsf{X} = [\mathsf{s_1}, \ldots, \mathsf{s_m}]\Lambda_\mathsf{m}\mathsf{P}^T, \tag{3.1}$$

which can be recognized as a compact representation of the *singular value decomposition (SVD)*[2],

$$\mathsf{X} = \mathsf{S}\begin{bmatrix} \Lambda_\mathsf{m} \\ 0 \end{bmatrix}\mathsf{P}^T. \tag{3.2}$$

We will now prove that the projection on the r first principal components is the optimal rank r orthogonal projection in the sense of maximizing the variance of the projected data. To start with, we will need to derive some properties of orthogonal projection matrices.

Definition 1. *The set of rank r orthogonal projections is the set of matrices*

$$\mathcal{P}_r = \{\Pi \in \mathbb{R}^{n\times n}; \Pi^2 = \Pi, \Pi^T = \Pi, \mathit{rank}(\Pi) = r\}$$

Let $\Pi \in \mathcal{P}_r$. Since Π is real and symmetric it can be diagonalized by an orthogonal matrix $\mathsf{U} = [\mathsf{u_1} \ldots \mathsf{u_n}]$ with the eigenvectors of Π as column elements. For Π, the following lemma holds:

Lemma 2. *If $\Pi = [\pi_1, \ldots, \pi_\mathsf{n}]$ is an orthogonal projection of rank $r \leq n$ in $\mathbb{R}^n$, then $|\pi_\mathsf{k}| \leq 1$ for all $k = 1, \ldots, n$ and $\sum_{k=1}^n |\pi_\mathsf{k}|^2 = r$.*

Proof. Let d_k, $k = 1, \ldots, n$ and $\mathsf{u_k}$, $k = 1, \ldots, n$ be the eigenvalues and eigenvectors of Π, respectively. Since $\Pi^2 = \Pi$, we have that

$$\Pi^2\mathsf{u_k} = \Pi\mathsf{u_k} \Rightarrow d_k^2\mathsf{u_k} = d_k\mathsf{u_k} \Rightarrow d_k = 1 \text{ or } d_k = 0 \qquad \text{for all } k = 1, \ldots, n.$$

Since the number of non-zero eigenvalues is $\mathrm{rank}(\Pi) = r$ it follows that

$$d_k = \begin{cases} 1 & k = 1, \ldots, r \\ 0 & \text{otherwise} \end{cases}$$

Now,

$$\sum_{k=1}^n |\pi_\mathsf{k}|^2 = \mathrm{tr}(\Pi^T\Pi) = \mathrm{tr}(\Pi) = \sum_{k=1}^n d_k = r.$$

To show that $|\pi_\mathsf{k}| \leq 1$, we first note that

$$\Pi = \mathsf{U}^T\mathrm{diag}(d_1, \ldots, d_n)\mathsf{U} = \{\sum_{i=1}^r \mathsf{U_{ij}}\mathsf{U_{ik}}\}_{jk}.$$

[2]This identification provides a numerically stable way of computing principal components and, in fact, the method of PCA is the same thing as SVD.

From the orthonormality of U it follows that $|\pi_\mathsf{k}| = \Pi_{\mathsf{kk}} = \sum_{i=1}^{r} u_{ik}^2 \leq 1$.

□

Having concluded this, we also need to define variance.

Definition 2. *The variance, Var*(X) *of* X *is the trace of its covariance matrix* C.

The trace of a matrix equals the sum of its eigenvalues so $\mathrm{Var}(\mathsf{X}) = \sum_{k=1}^{n} \lambda_k^2$.

Now, apply a rank r orthogonal projection Π to X, i.e., $\mathsf{X} \mapsto \Pi\mathsf{X}$. The optimal variance of $\Pi\mathsf{X}$ over all $\Pi \in \mathcal{P}_r$ is then

$$\sup_{\Pi\in\mathcal{P}_r} \mathrm{tr}(\Pi\mathsf{X}\mathsf{X}^T\Pi^T) = \sup_{\Pi\in\mathcal{P}_r} \mathrm{tr}(\Pi\mathsf{S}\Lambda^2\mathsf{S}^T\Pi^T) = \sup_{\Pi\in\mathcal{P}_r} \mathrm{tr}(\mathsf{S}^T\Pi\mathsf{S}\Lambda^2\mathsf{S}^T\Pi^T\mathsf{S}),$$

where we have used ,in the first step, Eq. (3.2), and in the second step, the fact that the trace is invariant under similarity transformations, $\mathsf{A} \mapsto \mathsf{S}^{-1}\mathsf{A}\mathsf{S}$. Noting that $\Pi \in \mathcal{P}_r$ iff $\mathsf{S}\Pi\mathsf{S}^T \in \mathcal{P}_r$, we conclude that

$$\sup_{\Pi\in\mathcal{P}_r} \mathrm{tr}(\mathsf{S}^T\Pi\mathsf{S}\Lambda^2\mathsf{S}^T\Pi^T\mathsf{S}) = \sup_{\Pi\in\mathcal{P}_r} \mathrm{tr}(\Pi\Lambda^2\Pi^T) = \sup_{\Pi\in\mathcal{P}_r} \sum_{k=1}^{n} \lambda_k^2 |\pi_\mathsf{k}|^2.$$

Lemma 2 gives an upper bound on $\mathrm{Var}(\Pi\mathsf{X})$:

$$\sum_{k=1}^{n} \lambda_k^2 |\pi_\mathsf{k}|^2 \leq \sum_{k=1}^{r} \lambda_k^2.$$

Furthermore, it is clear that equality is obtained by choosing $|\pi_\mathsf{k}|^2 = 1$ for $k = 1, \ldots, r$, and $|\pi_\mathsf{k}|^2 = 0$ otherwise. This optimum was obtained by making two transformations of the argument Π. Using the relation $|\pi_\mathsf{k}|^2 = \Pi_{\mathsf{kk}}$, and inverting these transformations, the optimal projection is

$$\Pi_{\mathsf{opt}} = \mathsf{S}\mathsf{S}^T \begin{bmatrix} \mathsf{I_r} & 0 \\ 0 & 0 \end{bmatrix} \mathsf{S} = \begin{bmatrix} \mathsf{I_r} & 0 \\ 0 & 0 \end{bmatrix} \mathsf{S},$$

where $\mathsf{I_r}$ is the $r \times r$ unit matrix. The optimal r-dimensional configuration is then

$$\mathsf{Z} := \mathsf{P}\Pi_{\mathrm{opt}}\mathsf{P}^T\mathsf{X} = \mathsf{P}\Pi_{\mathrm{opt}} \begin{bmatrix} \Lambda \\ 0 \end{bmatrix} \mathsf{S}^T = \mathsf{P} \begin{bmatrix} \Lambda_\mathsf{r} \\ 0 \end{bmatrix} \mathsf{S}^T,$$

where Λ_r is the diagonal matrix with diagonal elements λ_i if $i \leq r$ and 0 otherwise. In other words, Z is obtained from the singular value decomposition by replacing singular values $r+1$ to m with zeros. Thus, we have proved the following theorem:

Theorem 1. *Let* $\mathsf{X} \in \mathbb{R}^{n\times m}$ *be the data matrix of a set of* m *points in* $\mathbb{R}^n$. *The mapping*

$$\Psi_{PCA} : \mathsf{X} \mapsto \mathsf{P} \begin{bmatrix} \Lambda_r \\ 0 \end{bmatrix} S^T$$

is the orthogonal projection that maximizes the variance of the projection image configuration.

What is the relevance of looking for optimally variance preserving projections? We will further characterize the properties of the PCA projection with two corollaries.

Corollary 1. *The rank r principal component projection Ψ_{PCA} provides the optimal rank r approximation to X in squared Frobenius norm, i.e., it minimizes*

$$||\mathsf{X} - \Pi\mathsf{X}||_F^2,$$

over all $\Pi \in \mathcal{P}_r$.

Proof. First note that

$$\begin{aligned}
&||\mathsf{X} - \Pi\mathsf{X}||_{\mathrm{F}}^2 - ||\mathsf{X}||_{\mathrm{F}}^2 + ||\Pi\mathsf{X}||_{\mathrm{F}}^2 = \\
&= \mathrm{tr}(\mathsf{X}\mathsf{X}^T - \mathsf{X}\mathsf{X}^T\Pi^T - \Pi\mathsf{X}\mathsf{X}^T + \Pi\mathsf{X}\mathsf{X}^T\Pi^T) - \mathrm{tr}(\mathsf{X}\mathsf{X}^T) + \mathrm{tr}(\Pi\mathsf{X}\mathsf{X}^T\Pi^T) = \\
&= \mathrm{tr}(\Pi\mathsf{X}\mathsf{X}^T(\mathsf{I} - \Pi^T)) - \mathrm{tr}((\mathsf{I} - \Pi)\mathsf{X}\mathsf{X}^T\Pi^T) = 0,
\end{aligned}$$

since $\mathrm{tr}(\mathsf{A}) = \mathrm{tr}(\mathsf{A}^T)$. This results gives

$$\begin{aligned}
||\mathsf{X} - \Pi\mathsf{X}||_{\mathrm{F}}^2 &= ||\mathsf{X}||_{\mathrm{F}}^2 - ||\Pi\mathsf{X}||_{\mathrm{F}}^2 \\
&= ||\mathsf{X}||_{\mathrm{F}}^2 - ||\mathsf{X}^T\Pi^T||_{\mathrm{F}}^2 \\
&= ||\mathsf{X}||_{\mathrm{F}}^2 - \mathrm{tr}(\mathsf{X}^T\Pi^T\Pi\mathsf{X}) \\
&= ||\mathsf{X}||_{\mathrm{F}}^2 - \mathrm{tr}(\Pi\mathsf{X}\mathsf{X}^T\Pi^T) \\
&= ||\mathsf{X}||_{\mathrm{F}}^2 - \mathrm{Var}(\Pi\mathsf{X}),
\end{aligned}$$

so the optimum is obtained once again by maximizing the projected variance, and following Theorem 1 this is obtained by the principal component projection. □

Corollary 2. *The rank r principal component projection Ψ_{PCA} provides the optimal rank r approximation to X in the sense of minimizing the difference of sums of squared distances, i.e., it minimizes*

$$\sum_{i,j} ||\mathsf{x}_i - \mathsf{x}_j||^2 - ||\Pi\mathsf{x}_i - \Pi\mathsf{x}_j||^2,$$

over all $\Pi \in \mathcal{P}_r$.

Proof. We may rewrite the expression as

$$\begin{aligned}
\sum_{i,j} ||\mathsf{x}_i - \mathsf{x}_j||^2 - ||\Pi\mathsf{x}_i - \Pi\mathsf{x}_j||^2 &= \sum_{i,j} ||\mathsf{x}_i - \mathsf{x}_j||^2 - \mathsf{x}_i{}^T\Pi^T\Pi\mathsf{x}_i - \mathsf{x}_j{}^T\Pi^T\Pi\mathsf{x}_j + 2\mathsf{x}_i{}^T\Pi^T\Pi\mathsf{x}_j \\
&= \sum_{i,j} ||\mathsf{x}_i - \mathsf{x}_j||^2 - 2m\mathrm{tr}(\mathsf{X}^T\Pi^T\Pi\mathsf{X}) = \sum_{i,j} ||\mathsf{x}_i - \mathsf{x}_j||^2 - 2m\mathrm{tr}(\Pi\mathsf{X}\mathsf{X}^T\Pi^T),
\end{aligned}$$

where we have used that

$$\sum_{i,j} \mathsf{x_i}^T \Pi^T \Pi \mathsf{x_j} = \sum_i \mathsf{x_i}^T \Pi^T \Pi \sum_j \mathsf{x_j} = 0,$$

since $\sum_j \mathsf{x_j} = 0$. The minimal distance discrepancy is therefore given by an optimally variance preserving projection, which, by Theorem 1, is obtained by the principal component projection. □

Recall Example 4 in Section 3.1, which illustrated a case where data was scattered along a curved surface in $\mathbb{R}^3$. We noted that, using PCA for dimensionality reduction, this surface will not be optimally recovered. To make this a bit more precise, Corollary 2 implies that if $\mathcal{X}$ is a p-dimensional affine subspace of $\mathbb{R}^n$ then the p-dimensional PCA projection is an accurate reconstruction of $\mathcal{X}$ in the sense that $|z_i - z_j| = d_{\mathcal{X}}(\mathsf{x_i}, \mathsf{x_j})$, for all $i, j = 1, \ldots, m$, where $d_{\mathcal{X}}$ is the geodesic distance on $\mathcal{X}$. Furthermore, if $d \leq p$ the d-dimensional PCA projection is an optimal d-dimensional reconstruction of $\mathcal{X}$ in the sense of minimizing $\sum_{i,j} |d_{\mathcal{X}}(\mathsf{x_i}, \mathsf{x_j})|^2 - |\mathsf{z_i} - \mathsf{z_j}|^2$. If $\mathcal{X}$ is not affine then generally $d_{\mathcal{X}}(\mathsf{x_i}, \mathsf{x_j}) \neq |\mathsf{x_i} - \mathsf{x_j}|$ and hence we cannot guarantee the same kind of optimality of PCA as dimensionality reduction method.

3.3.2 Remarks

Scaling of variables. The fact that variance is the projection index of PCA makes it dependent on the scale of the variables. For example, if one group of variables measure length and another group of variables measure volume, then their magnitudes and their natural scale of variances will differ. The most common solution to this problem is to rescale all variables to unit variance. A problem with this approach, however, is that low magnitude noise is scaled up to the same magnitude as real signals. *Pareto scaling* is another solution, implementing a compromise between unit variance scaling and no scaling by dividing the variables by the square root of their standard deviation. If variables are known to be divided into a number of distinct categories one can make use of *block scaling* where variables are rescaled so that all groups have equal total variance.

Partial Least Squares. A method related to PCA is *Partial Least Squares (PLS)* [Wold, 1966], which is used when the same objects are described by two alternative data matrices, for example, gene expression data and values of chosen clinical variables. Let us, for a moment, abandon the notation of Section 3.2 and denote these two data sets by $\{\mathsf{x_i}\}$ and $\{\mathsf{y_i}\}$, with $i = 1, \ldots, m$ and call the corresponding data matrices X and Y respectively. The aim of PLS is to find projections $\Pi_{\mathsf{x}}\mathsf{X}$ and $\Pi_{\mathsf{y}}\mathsf{Y}$ such that the sample covariance between the two projections is maximized. This is obtained by the solutions of the eigenvalue problem

$$\begin{bmatrix} 0 & \mathsf{X}\mathsf{Y}^T \\ \mathsf{Y}\mathsf{X}^T & 0 \end{bmatrix} \begin{bmatrix} \mathsf{u_x} \\ \mathsf{u_y} \end{bmatrix} = \lambda \begin{bmatrix} \mathsf{u_x} \\ \mathsf{u_y} \end{bmatrix}.$$

Canonical Correlation Analysis Yet another method for the case with double data matrices X,Y is *Canonical Correlation Analysis (CCA)* [Hotelling, 1936]. In CCA it is the

correlation between the projections $\Pi_x \mathbf{X}$ and $\Pi_y \mathbf{Y}$ that is maximized, rather than the covariance as in PLS. The CCA projections are found by solving the generalized eigenvalue problem

$$\begin{bmatrix} 0 & \mathbf{X}\mathbf{Y}^T \\ \mathbf{Y}\mathbf{X}^T & 0 \end{bmatrix} \begin{bmatrix} u_x \\ u_y \end{bmatrix} = \lambda \begin{bmatrix} \mathbf{X}\mathbf{X}^T & 0 \\ 0 & \mathbf{Y}\mathbf{Y}^T \end{bmatrix} \begin{bmatrix} u_x \\ u_y \end{bmatrix}.$$

Since CCA works with dimensionless correlations it is more sensitive to over-fitting than PLS and therefore it is sometimes advised to use *regularized canonical correlation analysis* (RCCA). The equation to be solved is then

$$\begin{bmatrix} 0 & \mathbf{X}\mathbf{Y}^T \\ \mathbf{Y}\mathbf{X}^T & 0 \end{bmatrix} \begin{bmatrix} u_x \\ u_y \end{bmatrix} = \lambda \begin{bmatrix} \mathbf{X}\mathbf{X}^T + \gamma \mathbf{I} & 0 \\ 0 & \mathbf{Y}\mathbf{Y}^T + \gamma \mathbf{I} \end{bmatrix} \begin{bmatrix} u_x \\ u_y \end{bmatrix}.$$

This implements the heuristic of preferring high-variance before low-variance projections. Note that $\gamma \to \infty$ corresponds to PLS (after a re-scaling of the eigenvalues).

3.4 Kernel PCA

Consider Example 4 in Section 3.1 where measurements from a gravitational system are scattered on the surface $f = m/r^2$ in $\mathbb{R}^3$. As noted, PCA does not handle this kind of nonlinearities, since no orthogonal projection will capture all the variance. One way to get around this problem is to map the data into another space where variable dependencies are linear. Hence, transforming the data according to $\Psi : [m, r, f] \mapsto [\ln m, \ln r, \ln f]$ makes the objects lie on a plane ($\ln f = \ln m - 2\ln r$), and standard PCA may be applied to recover the two underlying degrees of freedom.

The idea of applying some nonlinear mapping into a linearizing feature space has been successfully exploited in classification, where the mapping of input data into feature space can make sample classes linearly separable and thereby easier to learn by a classifier. In particular, the use of *kernel support vector machines (SVMs)* [Boser et al., 1992] has had a large impact. Support vector machines and other *kernel methods* never explicitly carry out the mapping $\Psi : \mathcal{X} \to \mathcal{Z}$ of input data. Instead, data in input space are mapped onto scalar products in feature space $\mathcal{Z}$ by a *kernel function*, k. This saves computations, both since $\Psi(x_i)$ does not have to be evaluated, and since the subsequent calculations in $\mathcal{Z}$ are restricted to the linear span of the image of $\{x_i\}$ under Ψ. This approach is often called the *kernel trick* and can be applied to any algorithm that can be expressed solely using scalar products. As will be shown below, PCA is such an algorithm and this lead Schölkopf et al. [1998] to propose *kernel PCA*. Before describing kernel PCA in more detail we will briefly discuss some general properties of kernel methods.

3.4.1 Characterizing valid kernels

An obvious question that arises is which functions that are valid kernel functions. Before investigating this more closely we need some definitions. First, let $\mathcal{Z}$ be a linear space with an inner product $\langle \cdot, \cdot \rangle$.

Definition 3. *A* kernel *is a function* $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$*, such that for all* $x, x' \in \mathcal{X}$*,*

$$k(x, x') = \langle \Psi(x), \Psi(x') \rangle,$$

where Ψ *is a mapping from* $\mathcal{X}$ *to* $\mathcal{Z}$*.*

Definition 4. *The* Gram matrix G *of a set of vectors* $\{z_1, \ldots, z_m\}$ *is the* $m \times m$ *matrix with elements* $\mathsf{G}_{ij} = \langle z_i, z_j \rangle$*.*

Definition 5. *A* kernel matrix K *is the Gram matrix in a feature space* $\mathcal{Z}$ *corresponding to a kernel function* k*, i.e.,* $\mathsf{K}_{ij} = k(x_i, x_j)$*.*

It is clear that $k(x_i, x_j)$ needs to be symmetric, $k(x_i, x_j) = k(x_j, x_i)$ and, necessarily, it has to fulfill the Cauchy-Schwarz inequality, $k(x_i, x_j)^2 \leq k(x_i, x_i)k(x_j, x_j)$. Additionally, it is required that

$$\int_{\mathcal{X} \times \mathcal{X}} k(x, x') f(x) f(x') dx dx' \geq 0, \tag{3.3}$$

for all $f \in L_2(\mathcal{X})$ [Cristianini and Shawe-Taylor, 2000]. If this is the case, *Mercer's theorem* states that $k(x, x')$ can be expressed as

$$k(x, x') = \sum_{j=1}^{\infty} \lambda_j \Psi_j(x) \Psi_j(x'), \tag{3.4}$$

where (λ_j, Ψ_j) are the eigenvalues and eigenfunctions of the operator

$$(T_k f)(\cdot) = \int_{\mathcal{X}} k(\cdot, x) f(x) dx. \tag{3.5}$$

The eigenvalues are all non-negative and the eigenfunctions are assumed to be normalized so that $||\Psi_j||_{L_2} = 1$. Note that we may write $\hat{\Psi}_j = \lambda_j \Psi_j$ and thus $k(x, x')$ is the inner product in $\Psi(\mathcal{X})$. In the finite-dimensional case, the Mercer condition becomes a condition of positive semi-definiteness of K. This can be seen by letting f tend to a weighted sums of delta functions at each x_i which turns (3.3) into

$$\mathsf{v}^T \mathsf{K} \mathsf{v} \geq 0, \tag{3.6}$$

where v is the vector containing the values of f at $x_1, \ldots, x_m$. This allows us to formulate the following proposition (see e.g. [Cristianini and Shawe-Taylor, 2000]):

Proposition 1. *Let* $\mathcal{X}$ *be a finite input space with* $k(x, x')$ *a symmetric function on* $\mathcal{X} \times \mathcal{X}$*. Then* k *is a kernel function if and only if the kernel matrix* K *is positive semi-definite.*

3.4.2 Standard kernels

Thus far, we have established criteria to determine what is a *valid* kernel. We have not discussed what is a *suitable* kernel. Naturally, this question is more difficult to answer. Normally, we do not have any prior knowledge about which mapping Ψ will linearize the data, like we had for the gravity data in Example 4. Instead, it is common to use some standard kernel and hope that it will map data into a space where it is easier to work with.

In particular two kernel functions have been widely adopted in different kernel methods. These are the *Gaussian kernel*, $k(x,x') = e^{-||x-x'||^2/\sigma^2}$ and the (d order) *polynomial kernel*, $k(x,x') = (\langle x,x'\rangle + \sigma)^d$, where, in both cases, σ is a continuous parameter. Here we will briefly note on some illustrative properties of the Gaussian kernel.

The (squared) distance in Gaussian kernel feature space can be written as

$$||\Psi(x) - \Psi(x')||^2 = k(x,x) - 2k(x,x') + k(x',x') = 1 - 2e^{-||x-x'||^2/\sigma^2} + 1 =$$
$$= 2 - 2(1 - \frac{||x-x'||^2}{\sigma^2} + \frac{||x-x'||^4}{2\sigma^4} - \ldots) = \frac{||x-x'||^2}{\sigma^2}(1 - \frac{||x-x'||^2}{2\sigma^2} + \ldots).$$

Thus, when $||x-x'||$ is small compared to σ the feature space distance is proportional to the Euclidean distance in input space. If, on the other hand, $||x-x'||$ is large, $k(x,x') \to 0$ so $||\Psi(x) - \Psi(x')|| \to \sqrt{2}$, for all x,x'. Put in another way, varying σ between 0 and ∞ makes the feature space configuration travel between an m-point simplex and a scaled version of the input configuration.

3.4.3 Kernelized PCA

In order to derive a kernel version of PCA we need to reformulate the problem in terms of inner products. First, let $\mathsf{H} = \mathsf{I} - \frac{1}{m}\mathsf{1}\mathsf{1}^T$ be the $m \times m$ *centering matrix*. Next, recall from Section 3.3.1 that the central problem in PCA was to solve the eigenvalue problem

$$\mathsf{C}\mathsf{s}_\mathsf{k} = \mu_k \mathsf{s}_\mathsf{k} \qquad k = 1,2,\ldots,n, \tag{3.7}$$

where $\mathsf{C} = \mathsf{H}\mathsf{X}\mathsf{X}^T\mathsf{H}^T$ is the covariance matrix. What is required is thus to show how this can be reformulated as an eigenvalue problem for the dual matrix of C, i.e., $\mathsf{G} = \mathsf{H}^T\mathsf{X}^T\mathsf{X}\mathsf{H}$, the centered Gram matrix. We will now assume that X is centered so that $\mathsf{C} = \mathsf{X}\mathsf{X}^T$ and $\mathsf{G} = \mathsf{X}^T\mathsf{X}$.

First, note that

$$\mathsf{s}_\mathsf{k} = \mathsf{X}\frac{\mathsf{X}^T\mathsf{s}_\mathsf{k}}{\mu_k},$$

why we can replace s_k with $\mathsf{X}\mathsf{p}_\mathsf{k}$, where $\mathsf{p}_\mathsf{k} = \mathsf{X}^T\mathsf{s}_\mathsf{k}/\mu_k$ are the *dual variables*. Inserting this into (3.7) we get:

$$\mathsf{X}\mathsf{X}^T\mathsf{X}\mathsf{p}_\mathsf{k} = \mu_k\mathsf{X}\mathsf{p}_\mathsf{k} \tag{3.8}$$
$$\mathsf{X}^T\mathsf{X}\mathsf{X}^T\mathsf{X}\mathsf{p}_\mathsf{k} = \mu_k\mathsf{X}^T\mathsf{X}\mathsf{p}_\mathsf{k} \tag{3.9}$$
$$\mathsf{G}^2\mathsf{p}_\mathsf{k} = \mu_k\mathsf{G}\mathsf{p}_\mathsf{k}. \tag{3.10}$$

When G is of full rank this directly leads to

$$\mathsf{G}\mathsf{p}_\mathsf{k} = \mu_k\mathsf{p}_\mathsf{k}. \tag{3.11}$$

When G is rank deficient, then $\alpha + \alpha_0$, where α is a solution of (3.11) and α_0 is in the null space of G, is a solution of (3.10), but generally not of (3.11). The corresponding primal

variable in this case is $s = \mathsf{X}(\alpha + \alpha_0)$, but since $G\alpha_0 = 0$ we have that $\mathsf{X}\alpha_0 = 0$ and thus $s = \mathsf{X}\alpha$. Hence, all relevant solutions of (3.10) are those of (3.11) also in the case when G is rank deficient.

With this formulation of PCA, we may apply the kernel trick and replace G with K and solve

$$\mathsf{K}\mathsf{p}_\mathsf{k} = \mu_k \mathsf{p}_\mathsf{k}, \tag{3.12}$$

to find principal components in feature space. The kernel matrix needs to fulfill the conditions in Proposition 1 and furthermore we have assumed that data is centered so that $\mathsf{K}\mathsf{1} = \mathsf{0}$. If this is not the case, we may center the data in feature space by transforming $\mathsf{K} := \mathsf{HKH}$. Recall from 3.3.1 that we chose the set of primary eigenvectors s_i to be an orthonormal set. The same should hold in $\mathcal{Z}$ so with the primal eigenvectors in $\mathcal{Z}$ denoted by v_i we should have $\mathsf{v}_\mathsf{i}{}^T\mathsf{v}_\mathsf{i} = \mathsf{p}_\mathsf{i}{}^T\mathsf{X}^T\mathsf{X}\mathsf{p}_\mathsf{i} = \mu_i \mathsf{p}_\mathsf{i}\mathsf{p}_\mathsf{i} = 1$, why the dual variables p_i should be normalized to have length $1/\sqrt{\mu_i}$.

In order to attain a lower-dimensional kernel PCA reconstruction we need to project data points in feature space onto the v_k. The coordinate of a point $\Psi(\mathsf{x}_\mathsf{j})$ in the projection onto a primal eigenvector v_k can be calculated as

$$\begin{aligned}\langle \mathsf{v}_\mathsf{k}, \Psi(\mathsf{x}_\mathsf{j})\rangle &= \langle\, \Psi(\mathsf{X})\mathsf{p}_\mathsf{k}, \Psi(\mathsf{x}_\mathsf{j})\rangle = \\ &= \langle \sum_i (\mathsf{p}_\mathsf{k})_i \Psi(\mathsf{x}_\mathsf{i}), \Psi(\mathsf{x}_\mathsf{j})\rangle = \\ &= \sum_i (\mathsf{p}_\mathsf{k})_i k(x_i, x_j) = \mu_k (\mathsf{p}_\mathsf{k})_j,\end{aligned}$$

where $(\mathsf{p}_\mathsf{k})_i$ denotes the ith element of the vector p_k, and where the last step makes use of (3.12). Similarly the projection in feature space of a general point x can be found as

$$\langle \mathsf{v}_\mathsf{k}, \Psi(\mathsf{x})\rangle = \sum_i (\mathsf{p}_\mathsf{k})_i k(x_i, x). \tag{3.13}$$

Hence, new data samples, not used in the kernel PCA, can be projected into the lower-dimensional representation without recomputing the kernel eigenvectors. It has been noted (e.g. [Burges, 2005]) that this is equivalent to using the Nyström method to approximate the full eigenfunctions at the novel data point.

Kernel PCA projects data onto lower-dimensional subspaces in feature space. In many applications, it is of interest to map these projected data back into input space in order to interpret their meaning in the input space context. An arbitrary point z in the linear span of $\{\Psi(x_i)\}, i = 1, \ldots, m$ is not guaranteed to have a pre-image in $\mathcal{X}$ [Schölkopf et al., 1999], so the *pre-image problem* is somewhat difficult to solve. One approach is to minimize $||z - \Psi(x)||^2$ with respect to $x \in \mathcal{X}$ using some gradient method [Mika et al., 1999]. Since $z = \sum_i \beta_i \Psi(x_i)$ this loss function can be expressed in terms of the kernel function. Another suggestion is to use regression techniques to learn the mapping from $\mathcal{Z}$ to $\mathcal{X}$ using the pairs $(x_i, \Psi(x_i))$, $i = 1, \ldots, m$ [Bakir et al., 2004].

3.4.4 Remarks

Inefficiency of the Gaussian kernel for dimensionality reduction. The Gaussian kernel is a popular choice in support vector classification and, probably by merit of this, its use has spread over to kernel PCA applications. However, as pointed out in [Weinberger et al., 2004], it has a property that is not well suited for dimensionality reduction. If x and x' are far apart in input space, their inner product in feature space will be $k(x,x') = e^{-||x-x'||^2/\sigma^2} \approx 0$, meaning that $\Psi(x)$ and $\Psi(x')$ will be nearly orthogonal. Consequently, the dimensionality will increase rather than decrease and furthermore in a way that does not necessarily linearize the data.

3.4.5 Applications

In gene expression data analysis, kernel PCA appears relatively sparsely. One example, however, is [Pochet et al., 2004], where kernel PCA followed by Fisher discriminant analysis is applied to classify various cancer data. In fact, the authors find that using the Gaussian kernel in the dimensionality reduction yields poor performance. This might be explained by discussion above on the inefficiency of the Gaussian kernel for dimensionality reduction.

An interesting kernel application is found in [Vert and Kanehisa, 2003], where a kernelized version of canonical correlation analysis (cf. Section 3.3) is used based on two kernels; one computed from the expression data and one derived from a pathway database.

3.5 Multidimensional scaling

Multidimensional scaling (MDS) is a family of methods with the common aim to, given a matrix of dissimilarities between objects, construct a configuration of points $\{z_j\}$ in Euclidean space such that their interpoint distances, in some sense, well represent the dissimilarities. In this section we will denote the given dissimilarities between objects j,k by δ_{jk} and the corresponding reconstruction distances by d_{jk} according to what is standard in the MDS literature.

3.5.1 Metric MDS

An important special case of metric MDS is *classical MDS* or *principal coordinate analysis* [Torgerson, 1952], where the aim is to minimize $\sum_{j,k=1}^{m}(\delta_{jk}^2 - d_{jk}^2)$. From Corollary 2 we know that *if* $\delta_{jk} = |x_j - x_k|$ then an optimal configuration $\{z_i\}$ is found by projecting onto the principal components. In order to do this we need the covariance matrix, or alternatively the Gram matrix, as described in Section 3.4. We will now see how to transform a Euclidean distance matrix into a Gram matrix. Since we generally do not know if the given distance matrix really is Euclidean we also need to describe which matrices are valid Gram matrices, i.e., such that they may have been generated by a point configuration in a Euclidean space.

Let $\mathsf{G} = \mathsf{H}^T\mathsf{X}^T\mathsf{X}\mathsf{H}$ be the centered Gram matrix as in Section 3.4. We also define the squared Euclidean distance matrix as

$$\mathsf{T}_{jk} = |\mathsf{x}_j - \mathsf{x}_k|^2, \qquad j,k = 1,\dots,m.$$

Both G and T are invariant under Euclidean transformations so without loss of generality we may assume that the data is already mean centered, $\mathsf{XH} = \mathsf{X}$. The matrices T and G are related:

Theorem 2. *For all* $\{\mathsf{x}_1,\dots,\mathsf{x}_m\} \in \mathbb{R}^{n\times m}$ *we have that*

$$\mathsf{T}_{jk} = \mathsf{G}_{jj} + \mathsf{G}_{kk} - 2\mathsf{G}_{jk}, \tag{3.14}$$

and

$$\mathsf{G} = -\frac{1}{2}\mathsf{HTH}. \tag{3.15}$$

Proof. The equality (3.14) is simply the polarity condition. To prove the second equality we first note that

$$\sum_{j=1}^{m} \mathsf{G}_{jk} = 0 \quad \text{and} \quad \sum_{k=1}^{m} \mathsf{G}_{jk} = 0.$$

Therefore, summing over j in (3.14) yields

$$\sum_{j=1}^{m} \mathsf{T}_{jk} = m|\mathsf{x}_k|^2 + \sum_{j=1}^{m} |\mathsf{x}_j|^2, \tag{3.16}$$

which, summing over k, gives

$$\sum_{j,k=1}^{m} \mathsf{T}_{jk} = 2m \sum_{j=1}^{m} |\mathsf{x}_j|^2. \tag{3.17}$$

Combining these two equalities we get

$$|\mathsf{x}_k|^2 = \frac{1}{m}\sum_{j=1}^{m} \mathsf{T}_{jk} - \frac{1}{m}\sum_{j=1}^{m} |\mathsf{x}_j|^2 = \frac{1}{m}\sum_{j=1}^{m} \mathsf{T}_{jk} - \frac{1}{2m^2}\sum_{j,k=1}^{m} \mathsf{T}_{jk},$$

which we may plug into (3.14) to get

$$\mathsf{T}_{jk} = \frac{1}{m}\sum_{i=1}^{m}(\mathsf{T}_{ij} + \mathsf{T}_{ik}) - \frac{1}{m^2}\sum_{i,i'=1}^{m} \mathsf{T}_{ii'} - 2\mathsf{G}_{jk}.$$

Rearranging and putting this in matrix notation finally yields (3.15).

□

The above mapping from the distance matrix to the Gram matrix can be applied to all $m \times m$ matrices, but all $m \times m$ matrices are not the Euclidean distance matrix of some possible configuration of m points. To determine which distance matrices that are valid we may reapply Proposition 1, which says that G is the Gram matrix of some point configuration if and only if G is symmetric and positive semi-definite. If this condition is fulfilled Theorem 2 lets us conclude that T is the squared distance matrix of the same configuration.

If $\mathbf{G}$ fails these criteria the classical MDS solution might still be informative. For example, if $\mathbf{G}$ has a few negative eigenvalues with small magnitude the solution can be assumed to give a good approximation. In other cases, it would be advisable to turn to other MDS methods who do not assume that δ_{jk} are Euclidean.

In general *metric MDS* the aim is to construct a point configuration such that $d_{jk} \approx f(\delta_{jk})$ for some specified function f. This is done by minimizing some goodness-of-fit measure over the point configuration $\{z_i\}$ and, sometimes, the function f (given that f belongs to some specific parameterized family of functions). Many different goodness-of-fit measures have been proposed, for example the (normalized) *stress*

$$S(\{z_i\}, f) = \frac{\sum_{jk} w_{jk}(d(z_j, z_k) - f(\delta_{jk}))^2}{\sum_{jk} d(z_j, z_k)^2}, \tag{3.18}$$

where w_{jk} are weights that can be used to control which distances should be given the most importance in the fitting procedure. One well-known example is the method of *Sammon mapping* [Sammon Jr, 1969] which lets f be the identity and uses $w_{jk} = 1/\delta_{jk}$ in order to give more importance to the accurate representation of small distances.

Most cost functions give rise to optimization problems that has to be solved numerically. Furthermore, it has been noted that the problems are often non-convex, why algorithms risk getting stuck in local minima. Typically, the classical MDS solution is used as an initial value.[3]

The idea of transforming the dissimilarities $\delta_{jk} \mapsto f(\delta_{jk})$ reminds of the approach that is taken in kernel methods. In fact, in [Williams, 2002] it is shown that kernel PCA with isotropic kernels can be interpreted as a type of metric MDS.

3.5.2 Non-metric MDS

For many types of data the meaning of the magnitudes of the dissimilarities δ_{jk} is unclear. For this purpose, *non-metric MDS* or *ordinal scaling* [Kruskal, 1964] has been devised. There are different non-metric MDS algorithms but they have in common that they attempt to create reconstructions so that the rank ordering of d_{jk} is similar to that of δ_{jk}, i.e., that

$$\delta_{ij} \leq \delta_{kl} \iff d_{ij} \leq d_{kl},$$

is optimally true. A standard non-metric MDS algorithm adopts the stress (Eq. (3.18)) as cost function and lets f be computed in every step by means of monotone regression.

Non-metric MDS, as just described, puts less constraints on the reconstruction $\{z_i\}$ than metric MDS. However, in some cases it is relevant to put even less constraints on the relation between d_{jk} and δ_{jk}. For example, we may only have the reason to require that the two dissimilarity matrices are *local order equivalent*, i.e., that

$$\delta_{ij} < \delta_{ik} \iff d_{ij} \leq d_{ik},$$

as described in [Sibson, 1972].

[3] However, Malone et al. [2002] suggests another way of choosing the initial configuration by solving a convex optimization problem whose solution can be argued to be close to the global stress minima.

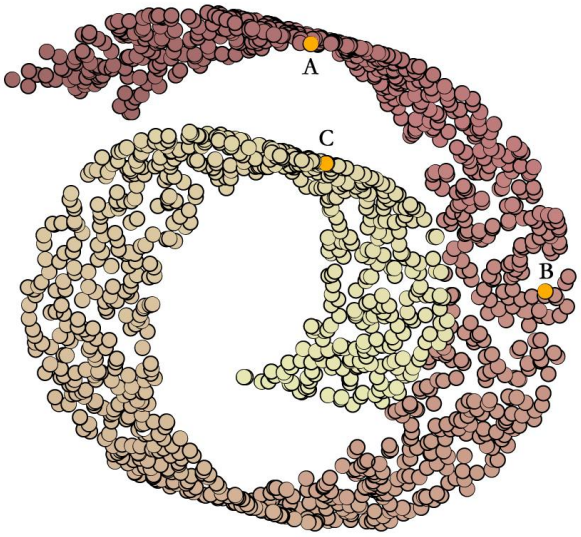

Figure 3.4: A Swiss roll manifold. The geodesic distance between A and B is shorter than that between A and C.

3.6 Isomap

In Section 3.5 the method of multidimensional scaling was introduced. Classical MDS takes as input Euclidean distances between points x_i on $\mathcal{X}$, and produces a lower-dimensional point configuration $\{z_i\}$ for which the pairwise Euclidean distances optimally match the Euclidean distances in input space. Now, if the input data lie on, or close to, a curved manifold $\mathcal{X}$, the distances between points z_i in the reconstruction will not correspond well with the distances between points y_i in the parameter space. This is the same problem as in Example 4. The following gives yet another illustration:

Example 5. *Suppose that data points are distributed uniformly on a rolled sheet, similar to a Swiss roll, as in Figure 3.4. It is natural to argue that the preferable distance measure between points is not the Euclidean distance, but the distance along the spiral. Consequently, in Figure 3.4, the distance between A and B should be considered shorter than that between A and A, which is not the case if we use the straight-line Euclidean distance.*

In other words, we would like to compute the *geodesic distance* between the points $\{x_i\} \in \mathcal{X}$. Once this is done we may use these distances as input to an MDS algorithm in order to create a lower dimensional representation of the data.

This section describes how approximations of geodesic distances are computed using the *Isomap* algorithm [Tenenbaum et al., 2000].

3.6.1 Approximating geodesic distances

The Isomap algorithm works as follows:

1. Construct an adjacency graph where the nodes $\{n_i\}$, $i = 1, \ldots, m$ represent the data points $\{x_i\}$ and two nodes n_i, n_j are connected if $n_i \in \mathcal{N}_j$ or $n_j \in \mathcal{N}_i$, where $\mathcal{N}_i$ is the set of points neighboring n_i. The neighborhood $\mathcal{N}_i$ can be defined either according to the ε-rule:

$$\mathcal{N}_i = \{n_k; \quad |x_k - x_i| < \varepsilon\},$$

for some $\varepsilon \in \mathbb{R}_+$, or according to the K-rule:

$$\mathcal{N}_i = \{n_k; \quad x_k \text{ is among the } K \text{ closest neighbors of } x_i\},$$

for some $K \in \mathbb{N}$. The graph edges are given weights $w_{ij} = |x_i - x_j|$.

2. For each pair of nodes n_i, n_j, find the shortest path in the graph between them. This can be done, for example using Dijkstra's algorithm. The length of this path is then the *approximate geodesic distance* $d_{Iso}(x_i, x_j)$ between x_i and x_j.

3. Compute reconstructed embedding coordinates by applying classical MDS to the approximate geodesic distances.

Steps 1 and 2 are the manifold learning steps of the method — it is here that the estimation of geodesic distances is performed. The dimensionality reduction is performed in step 3. The formulation above assumes that the adjacency graph is connected. However, if it is not, we would instead proceed by handling each connected subgraph separately.

So far we have loosely motivated the Isomap algorithm by an example. It remains to address questions concerning under which conditions it actually works.

For the manifold learning steps, the question is whether the approximate geodesic distances converge to the true geodesic distances as $m \to \infty$. As perhaps intuitively expected, they do, and the consistence proofs can be found in [Bernstein et al., 2000]. The convergence rate basically depends the curvature of $\mathcal{X}$.

Having stated that geodesic distances can be well approximated, the next question to answer is when the classical MDS step can create adequate lower-dimensional representations. In fact, it does so at least under the following *sufficient Isomap condition*:

> $\mathcal{X}$ is the image of an open convex subset $\mathcal{Y}$ of Euclidean space under an isometric mapping Φ.

In this case, the geodesic distances are identical to Euclidean distances in $\mathcal{Y}$ and classical MDS yields an adequate lower-dimensional approximation of the configuration in $\mathcal{Y}$. The isometry criterion implies that geodesics have the same length in $\mathcal{X}$ and $\mathcal{Y}$ while the convexity criterion assures that classical MDS works properly in $\mathcal{Y}$. If the criteria are not fulfilled, however, the MDS reconstructions may still yield informative results.

Example 6.

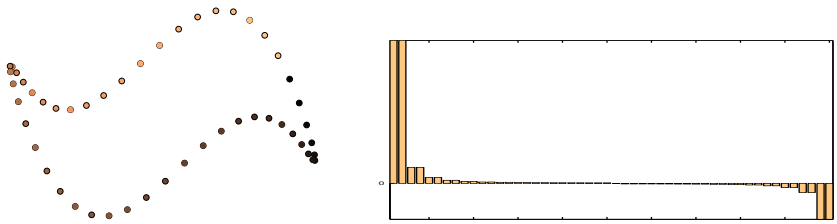

Figure 3.5: Classical MDS applied to geodesic distances on S^1. The left panel shows the three-dimensional reconstruction. The right panel shows a bar plot of the eigenvalues of the covariance matrix. Eigenvalues appear in consecutive pairs.

Let $\mathcal{X} = S^1$, i.e., a circle. Then there exists no combination of an open convex Euclidean subspace $\mathcal{Y}$ and an isometry Φ, such that $\mathcal{X} = \Phi(\mathcal{Y})$. Geodesic distances on $\mathcal{X}$ are the same as angular differences between vectors on the unit circle. Sampling a number of equidistant points on $\mathcal{X}$ and feeding the geodesic distance matrix into classical MDS yields a spectral decomposition of the covariance matrix where eigenvalues appear in pairs, thus describing a circle in every consecutive pair of dimensions (Figure 3.5).

Furthermore, note that even if the violations are such that dimensionality reduction using classical MDS can not be performed with meaningful results, the approximate geodesic distances are still valid and can be analyzed in other ways, for example, using metric or non-metric multidimensional scaling, or as inputs to clustering algorithms.

In reality, the density of data points will always be finite, and in many cases sparse, why it is important to study what kind of approximation errors appear when applying Isomap to finite data sets. One potential problem is that of *topological instability*, as pointed out in [Balasubramanian et al., 2002]. If the neighborhood parameter (ε or K) is chosen too large with respect to the density of data points and the curvature of the manifold, or if data is noisy or contains outliers, then shortcuts may appear in the adjacency graph, connecting geodesically distant domains of the manifold (see Figure 3.6 a). Such a shortcut inflicts big damage in the approximation of geodesic distances and accordingly disrupts the resulting embedding reconstruction. The problem of topological instability is the focus of Chapter 5, where a more robust method for approximation of geodesic distances is proposed.

Besides the topological instability which gives rise to global approximation errors, there is a local error effect appearing at finite data set sizes. Under such circumstances, holes appear in the adjacency graph due to random fluctuations in local density, as illustrated in Figure 3.6 b. For pairs of points on opposite sides of such holes the approximation error will become larger and consequently the holes grow in the resulting Isomap projection. In other words, the finite data density introduces small violations of the convexity condition. As a consequence, the projection might exhibit structures which are amplifications of random fluctuations in data density. This has been termed a 'Swiss cheese' effect [Lee et al., 2002], likely due to the popularity of the Swiss roll data in the manifold learning community. We will also refer to this as an over-clustering effect, and we note that it may make it appear as if there is structure in the data when in fact there is none. While the risk of introducing graph shortcuts grows with the neighborhood parameter value, the im-

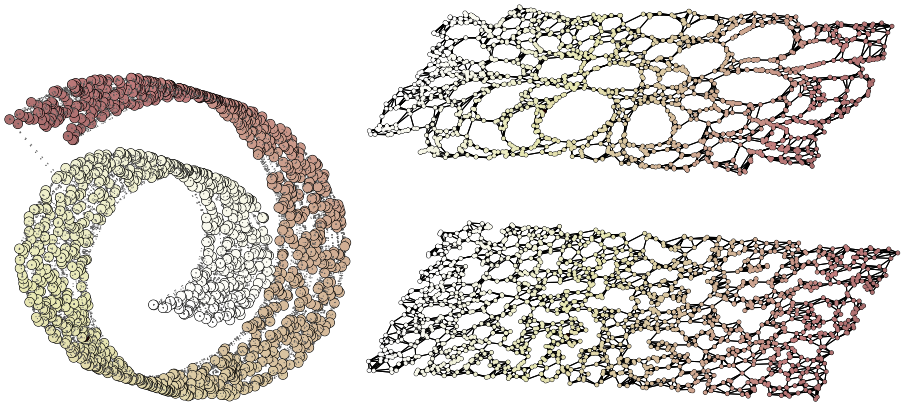

Figure 3.6: To the left is shown 1500 uniformly distributed points ($\hat{\mathcal{X}}_{1500}$) on a swiss roll manifold; the adjacency graph (K=10) contains two shortcuts. To the right, the corresponding 1500 points in $\hat{\mathcal{Y}}_{1500}$, uniformly distributed in the plane (lower) and the corresponding Isomap projection (upper), $\hat{\mathcal{Z}}_{1500}$. The adjacency graph (K=5) is drawn in the configurations. Random fluctuations in data density become amplified in the Isomap reconstruction.

pact of the over-clustering decreases with the parameter value, so, in practise, choosing the parameter values becomes a question of compromising between these two types of errors.

3.6.2 Remarks

Conformal Isomap. It is not difficult to construct an example where the isometry part in the Isomap condition fails to hold. For example, a patch, cut out from a sphere can not be described as the image of a Euclidean subspace under an isometric mapping. However, a rather simple modification of the algorithm extends the class of permitted mappings Φ in the sufficient condition to the strictly larger class of conformal mappings. The C-Isomap algorithm [de Silva and Tenenbaum, 2003] achieves this by re-weighting the edges in the adjacency graph by $|x_j - x_k|/\sqrt{M(x_i)M(x_j)}$, where $M(x_i)$ is the mean distance from x_i to its neighbors. If Φ is a conformal mapping, then a disk in $\mathcal{Y}$ of radius r, centered around y_i is mapped onto a disk of radius $r\varphi(y_i)$ where $\varphi(y_i)$ is the local magnifying factor. Hence, scaling distances between neighboring points x_i, x_j in $\mathcal{X}$ by $\sqrt{\varphi(y_i)\varphi(y_j)}^{-1}$ yields distances proportional to $d_{\mathcal{Y}}(y_i, y_j)$. Assuming that data is uniformly distributed in $\mathcal{Y}$, the radius r of a K-neighborhood around $y_i \in \mathcal{Y}$ does not depend on y_i, so $\varphi(y_i)$ may be estimated as being proportional to $M(x_i)$, thus explaining the exact form of the graph weights.

Landmark Isomap. The Isomap algorithm has two computational bottlenecks — the computation of shortest graph distances which is $O(m^2 \log m)$ [de Silva and Tenenbaum, 2003] using Dijkstra's algorithm and the solution of the MDS eigenvalue problem which

is $O(m^3)$ since the distance matrix is non-sparse. These problems can be relieved by implementing a landmark version of Isomap as described in [de Silva and Tenenbaum, 2003]. The L-Isomap algorithm chooses, by random, a number $m' << m$ of landmark points among $\hat{\mathcal{X}}_m$. By computing approximate geodesic distances between the landmarks only, embedding them in low dimension using MDS and triangulating the positions of the remaining points in the reconstructed embedding based on their original distances to the landmarks, the computational cost scales down considerably.

Vector Quantization. Lee et al. [2002] suggest that input data should be pre-processed using *vector quantization* in order to even out random density fluctuations and thereby reduce the impact of the over-clustering effect. This can be seen as an alternative way of choosing landmark points in L-Isomap.

Inefficiency of Isomap on periodic data. The convexity condition has some interesting consequences. For example it implies that Isomap is not well suited for periodical data, since the parameter space of such data is a non-convex subset of Euclidean space. A simple example in this respect is when $\mathcal{Y} = S^1$. Real life examples include cell cycle expression data and periodic image series such as images of rotating objects.

3.6.3 Applications

The application area in the original work by Tenenbaum et al. [2000] was in cognitive vision. By representing images as points in pixel space and applying Isomap to sets of images with clear underlying geometrical parametrization, such as wrist angle and degree of openness of a photographed hand (Figure 3.7), the authors demonstrated that the Isomap projections recovered the underlying parametrization. Motivated by these positive examples, Donoho and Grimes [2002] derived results showing that several types of image manifolds are indeed isometric to their parameterizations. The image manifolds were created by continuously articulating simple objects in the image plane, such as translations of a disk, rotations of a closed figure, articulations of a horizon, independent non-occluding motions of 'fingers' of a cartoon 'hand', and gestures of a cartoon 'face', with articulated features. Conversely, they showed that when two objects are articulated on the same image plane, while not being allowed to occlude each other, the convexity condition of Isomap is not fulfilled and Isomap fails to recover the underlying parametrization.

The application of Isomap to gene expression data is described in the paper [Nilsson et al., 2004] which is reproduced in Chapter 4. Subsequently, Isomap has also appeared in [Andersson et al., 2005a] and [Dawson et al., 2005].

Other applications of the Isomap algorithm includes neurophysiology [Laskaris and Ioannides, 2002], econometrics [Liou and Kuo, 2002] and shape analysis [Lim et al., 2003, Larsen, 2005].

3.7 Laplacian Eigenmaps

While Isomap attempts to infer the geodesic distances on the underlying manifold, *Laplacian Eigenmaps* [Belkin and Niyogi, 2003] focuses on other properties of $\mathcal{X}$. Using the

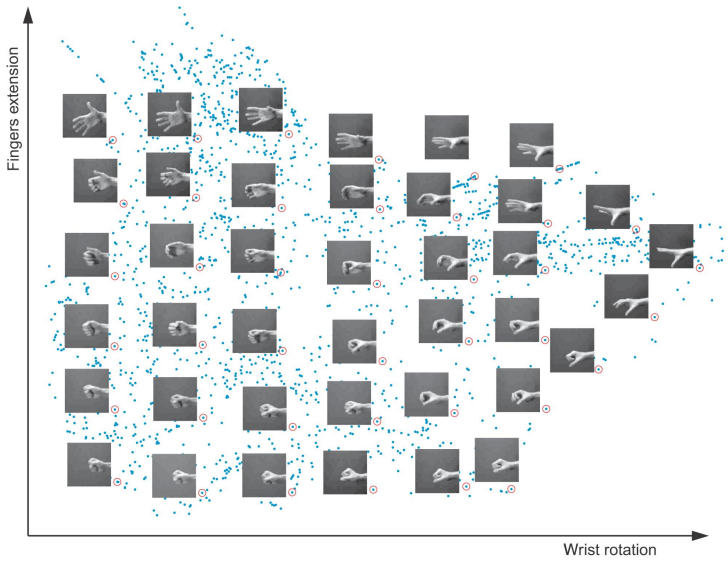

Figure 3.7: Isomap reconstruction of a set of images of a hand at different wrist angle and degree of openness. The figure originally appears in [Tenenbaum et al., 2000].

same construction of an adjacency graph, it seeks to approximate the Laplace–Beltrami operator on the manifold by the graph Laplacian matrix. The idea behind this is that an eigenfunction of the Laplace–Beltrami operator will have minimal average gradient on $\mathcal{X}$ and is therefore an appropriate mapping of samples into $\mathcal{Z}$, in the sense that it maps points that are close on $\mathcal{X}$ so that they are also close in $\mathcal{Z}$.

3.7.1 Finding gradient minimizing mappings

Consider the problem of finding a smooth, one-dimensional mapping $f : \mathcal{X} \longrightarrow \mathbb{R}$ from the data manifold to the real line, such that points that are nearby on $\mathcal{X}$ are as near as possible[4] on $\mathbb{R}$. It can be shown that for any points $x, x' \in \mathcal{X}$ it holds that

$$|f(x') - f(x)| \leq ||\nabla f(x)|| d_{\mathcal{X}}(x, x') + o(d_{\mathcal{X}}(x, x')). \tag{3.19}$$

Thus, $||\nabla f(x)||$ approximately measures how far apart f maps nearby points and if we are interested in minimizing this, a reasonable objective measure would be the functional

$$F(f) = \int_{\mathcal{X}} ||\nabla f||^2, \tag{3.20}$$

with the constraint $||f|| = 1$. Next, we note that for the Laplace-Beltrami operator $\mathcal{L}$, i.e., the manifold generalization of the Laplace operator on $\mathbb{R}^n$, defined as $\mathcal{L}f = -\text{div}\nabla f$, it holds that $\int_{\mathcal{X}} \langle V, \nabla f \rangle = -\int_{\mathcal{X}} \mathcal{L}(V) f$, for any vector field V, why

[4] under appropriate constraints

$$\int_{\mathcal{X}} ||\nabla f||^2 = \int_{\mathcal{X}} \mathcal{L}(f) f, \tag{3.21}$$

where we assume that $\mathcal{X}$ is without boundary. The minimizing function of $F(f)$ is an eigenfunction of $\mathcal{L}$.

As thus motivated, the geometrical entity of interest in this context is the Laplace-Beltrami operator. Recall that the Isomap algorithm constructed an adjacency graph and calculated graph geodesics as approximations to the underlying manifold geodesics. Laplacian Eigenmaps approximates the Laplace-Beltrami operator by the graph Laplacian of the adjacency graph.

To this end, let the matrix W be defined as $\mathsf{W}_{ij} := e^{-||x_i - x_j||^2/2\sigma^2}$ and let B be the degree matrix with entries $\mathsf{B}_{ii} = \sum_j \mathsf{W}_{ij}$. The *graph Laplacian* is then defined as $\mathsf{L} = \mathsf{B} - \mathsf{W}$. This matrix is positive semidefinite and symmetric. Furthermore it can be verified that $\mathbf{1} = [1, \ldots, 1]^T$ is an eigenvector with eigenvalue 0. Also, define the *normalized graph Laplacian* as $\mathsf{L}' = \mathsf{B}^{-1/2}\mathsf{L}\mathsf{B}^{-1/2}$.

Next, we will motivate why the normalized graph Laplacian, as defined above, can be seen as a discrete approximation of the Laplace-Beltrami operator [Belkin and Niyogi, 2003]. To this end, consider the heat equation,

$$(\frac{\partial}{\partial t} + \mathcal{L})u = 0, \tag{3.22}$$

where $u(x,t)$ is the heat distribution at time t and $u(x,0) = f(x)$ is the initial heat distribution. The solution of the heat equation is given by

$$u(x,t) = \int_{\mathcal{X}} H_t(x,y) f(y), \tag{3.23}$$

where $H_t(x,y)$ is the *heat kernel*. In geodesic coordinates, H_t is approximately the Gaussian:

$$H_t(x,y) = (4\pi t)^{-p/2} e^{-\frac{||x-y||^2}{4t}} (\varphi(x,y) + O(t)), \tag{3.24}$$

where $\varphi(x,y)$ is a smooth function with $\varphi(x,x) = 1$. By inserting (3.23) into the heat equation we may write

$$\mathcal{L}f(x) = -\mathcal{L}u(x,0) = -\left(\frac{\partial}{\partial t} \int_{\mathcal{X}} H_t(x,y) f(y)\right)_{t=0}. \tag{3.25}$$

In order to arrive at L' we will make three different approximations of (3.25). First, if t is small and x and y are close (relative to the curvature of $\mathcal{X}$),

$$H_t(x,y) \approx (4\pi t)^{-p/2} e^{-\frac{||x-y||^2}{4t}}. \tag{3.26}$$

Second, since $\lim_{t\to 0} \int_{\mathcal{X}} H_t(x,y) f(y) = f(x)$, we may, using the definition of the derivative, approximate the derivative at $t = 0$ as:

$$\mathcal{L}f(x) \approx \frac{1}{h}\left[f(x) - \int_{\mathcal{X}} (4\pi h)^{-p/2} e^{-\frac{||x-y||^2}{4h}} f(y)\right], \tag{3.27}$$

where h is some small time step. Finally, at high data densities ($m \to \infty$) we may find a discrete approximation of (3.27) as:

$$\mathcal{L}f(x_i) \approx \frac{1}{h}\left[f(x_i) - \frac{1}{m}(4\pi h)^{-p/2} \sum_{x_j \in \mathcal{N}_\varepsilon(x_i)} e^{-\frac{||x_j - x_i||^2}{4h}} f(x_i)\right], \tag{3.28}$$

where $\mathcal{N}_\varepsilon(x_i)$ is the set $\{x_j\}$ such that $||x_i - x_j|| < \varepsilon$. The manifold dimensionality p is generally unknown why we need to estimate the factor $\alpha_i := (4\pi h)^{-p/2}/m$ from the data. By inserting $f \equiv 1$ into (3.28) and rewriting we get

$$\alpha_i = \left(\sum_{x_j \in \mathcal{N}_\varepsilon(x_i)} e^{-\frac{||x_j - x_i||^2}{4h}}\right)^{-1}, \tag{3.29}$$

which we might recognize as $\mathsf{B}_{ii}{}^{-1}$. Hence, with W and B defined as above, but with $\sigma = h$, the eigenfunction problem $\mathcal{L}f = \lambda f$ approximately transforms into $(\mathsf{B} - \mathsf{W})f = \lambda \mathsf{B} f$ on the graph.

The derivation above contains a number of approximations, and strict matters of the consistency of the eigenvalues of L' as estimators of the eigenfunctions of the Laplace-Beltrami operator in the limits $m \to \infty$, $\sigma \to 0$ are still not fully investigated. The above arguments holds under the assumption that the density function with respect to the measure on $\mathcal{X}$ is uniform [Belkin, 2003]. For non-uniform densities, however, the asymptotic of L' is not the standard Laplace-Beltrami, but a weighted version of it [Lafon, 2004, Nadler et al., 2004, Hein et al., 2005]. Moreover, proving the convergence of the matrix to the right operator does not completely prove the consistency. What is needed is a proof of the convergence of its eigenvectors to the eigenfunctions of the operator. One step towards this is taken in [von Luxburg et al., 2004] where the convergence of the eigenvectors is shown in the limit $m \to \infty$ but with σ fixed.

Guided by the result that the standard normalized graph Laplacian does not generally have the desired asymptotic behavior a normalization of the weight matrix W was proposed in [Lafon, 2004] yielding the desired asymptotic guarantees for non-uniform densities as well;[5]

$$\tilde{\mathsf{W}} = \mathsf{B}^{-1}\mathsf{W}\mathsf{B}^{-1}. \tag{3.30}$$

In summary, the Laplacian Eigenmaps method works as described by the following algorithm:

[5] Nadler et al. [2004] extend this idea and present a parametric family of graph Laplacian normalizations including as asymptotic the backward Focker-Planck operator.

1. Construct an adjacency graph.
2. Calculate graph weights: $\mathsf{W}_{ij} = e^{-||x_i - x_j||^2/2\sigma^2}$.
3. Replace W by its normalized counterpart $\tilde{\mathsf{W}} = \mathsf{B}^{-1}\mathsf{W}\mathsf{B}^{-1}$. (This step is not part of the original algorithm but enables handling non-uniform underlying densities [Lafon, 2004].)
4. Compute eigenvalues and eigenvectors of the normalized graph Laplacian, i.e., solve

$$\mathsf{L}s = \lambda \mathsf{B}s. \tag{3.31}$$

Leave out the eigenvector s_0 corresponding to the zero eigenvalue λ_0, and form the matrix $\mathsf{S} = (s_1, \ldots, s_{m-1})$ from the (column) eigenvectors $s_1, \ldots, s_{m-1}$ corresponding to the descendent ordered eigenvalues $\lambda_1, \ldots, \lambda_{m-1}$. The p-dimensional Laplacian Eigenmaps embedding $\Psi(x_i)$ of sample i is then given by the first p elements in the i:th row of S, i.e., $\Psi(x_i) = [\psi_1(x_i), \ldots, \psi_p(x_i)]^T$ where $\psi_j(x_i) = \mathsf{S}_{ij}$.[6]

Apart from its role as an estimation of the eigenfunctions of the Laplace–Beltrami operator the mapping Ψ is optimal in the sense that it provides a configuration $\{z_j\}$, with a data matrix $\mathsf{Z} = [\mathsf{z}_1, \ldots, \mathsf{z}_m]$, that minimizes

$$\mathrm{tr}(\mathsf{Z}\mathsf{L}\mathsf{Z}^T) = \sum_{j,k} ||z_j - z_k||^2 \mathsf{W}_{jk},$$

under the constraint that $\mathsf{Z}\mathsf{B}\mathsf{Z}^T = \mathsf{I}$.

3.7.2 Remarks

Relation to PCA. Let $\sigma \to \infty$ and the set the neighborhood parameter $K = m - 1$, i.e., let all points be neighbors of each other. With these parameter values, Laplacian Eigenmaps is equivalent to PCA. The weight matrix becomes $\mathsf{W} = [\mathsf{1}\mathsf{1}\ldots\mathsf{1}]$ and the degree matrix $\mathsf{B} = m\mathsf{I}$. The dual problem to minimizing $\mathrm{tr}(\mathsf{Z}\mathsf{L}\mathsf{Z}^T)$ under the constraint $\mathsf{Z}\mathsf{B}\mathsf{Z}^T = \mathsf{I}$ is to maximize $\mathrm{tr}(\mathsf{Z}\mathsf{B}\mathsf{Z}^T)$ under the constraint $\mathsf{Z}\mathsf{L}\mathsf{Z}^T = c$, where c is a constant. Now, since $\mathsf{B} = m\mathsf{I}$ the desired solution is one that maximizes $\mathrm{tr}(\mathsf{Z}\mathsf{Z}^T)$ and this, as was shown in 3.3.1, is the principal component projection.

Relation to Fourier analysis. Consider the following example:

Example 7. *Let $\mathcal{X} = S^1$. The eigenfunctions of $\mathcal{L} = -\frac{d^2}{dx^2}$ are then given by*

$$u_k(x) = A\sin(2\pi k x + \alpha), \qquad k = 1, \ldots, \tag{3.32}$$

where A and α are real constants.

This reminds us of the link between the eigenfunctions of $\mathcal{L}$ and the basis functions of the Fourier expansion. In other words, the Laplacian Eigenmaps algorithm can be seen as a method of computing a Fourier basis for a given manifold. Interesting work is currently being done on *diffusion wavelets* — analogous algorithms for computing multiscale 'wavelet bases' on manifolds [Coifman et al., 2005].

Connection to spectral clustering. Spectral clustering methods solve the problem of partitioning a graph so that the flow, that is, the total weight of the edges, between partitions become as small as possible. Different variants of spectral clustering exists but they have in common that they solve eigenvalue problems similar to (3.31), and in the case of [Ng et al., 2002] exactly the same equation is solved. Hence, spectral clustering can be seen as clustering in the reconstructed embedding space given by Laplacian Eigenmaps.

Hessian Eigenmaps. *Hessian Eigenmaps* [Donoho and Grimes, 2003] is conceptually related to Laplacian Eigenmaps. While Laplacian Eigenmaps looks for functions on the manifold that have minimal average gradient, Hessian Eigenmaps looks for functions that have minimal average Hessian over $\mathcal{X}$. Briefly, the algorithm constructs an adjacency graph, estimates the Hessian matrix at each point, gathers the local Hessians in a single matrix, performs a spectral decomposition of this matrix, and extracts the eigenvectors corresponding to the smallest positive eigenvalues as coordinate functions for the reconstructed embedding. Hessian Eigenmaps has relatively broad theoretical asymptotical guarantees. If $\mathcal{X}$ is the image of an open connected subset $\mathcal{Y}$ of Euclidean space under a locally isometric mapping Φ, then the method will asymptotically reconstruct $\mathcal{Y}$ up to rigid motions. Hence, its global guarantees hold for a broader class of manifolds compared with Isomap. Laplacian Eigenmaps lacks such global guarantees altogether since it is a local method.

3.7.3 Applications

The Laplacian Eigenmaps method has been applied to, among other fields, neurophysiology[Brun et al., 2003] and face recognition [He et al., 2005].

In gene expression data analysis Laplacian Eigenmaps appears in [Venna and Kaski, 2005], where Laplacian Eigenmaps and other methods are applied to visualize large data banks consisting of multiple data sets.

3.8 Locally Linear Embedding

Isomap creates reconstructions based on the assumption that data was generated by an isometric mapping of points from a convex set, and that thus global distances are preserved by the mapping Φ. In contrast, the method of *Locally Linear Embedding (LLE)* Roweis and Saul [2000] makes the assumption that *local* structure is preserved while mapping from $\mathcal{Y}$ to $\mathcal{X}$. As a consequence, LLE can handle a larger class of mappings but the reconstructions can only be trusted to be locally correct. In other words, LLE, like Laplacian Eigenmaps, is an example of a local method, as defined in Section 3.2, while Isomap is a global method.

3.8.1 Finding locality preserving projections

The key idea in LLE is that if the neighborhood $\mathcal{N}_i$ of $x_i \in \mathcal{X}$ lies on a locally linear patch, x_i can be expressed as a linear combination of its neighbors. Now, the weights of this linear combination are invariant to translations, rotations and scalings of the neighborhood, so if Φ looks like such a transformation locally around y_i then the same weights can be used to construct y_i from its neighbors $\Phi^{-1}(\mathcal{N}_i)$.

The LLE algorithm goes as follows:

1. Detect neighborhoods $\mathcal{N}_i$.
2. Calculate neigborhood weights, i.e., find w_{ij} such that
$$\sum_i |x_i - \sum_{j \in \mathcal{N}_i} w_{ij} x_j|^2,$$
is minimized.
3. Calculate reconstructed embedding coordinates, i.e., find $\{z_i\}$ such that
$$\sum_i |z_i - \sum_{j \in \mathcal{N}_i} w_{ij} z_j|^2,$$
is minimized.

3.8.2 Applications

Shi and Lihui [2005] used Locally linear embedding for dimensionality reduction of microarray data and compared classification performance of an SVM trained on the lower-dimensional projection. Another application work appears in [Jain and Saul, 2004] where LLE is used for speech recognition.

3.9 Kernel formulations

In Section 3.4, we noted the difficulty of choosing appropriate explicit kernels and in particular we saw that the Gaussian kernel is not well suited for dimensionality reduction. Isomap, Laplacian Eigenmaps and LLE does not share this problem since here the mapping from $\mathcal{X}$ to $\mathcal{Z}$ is implicitly learned from data. As noted by Ham et al. [2004], these methods may be interpreted as kernel PCA with a kernel learned from data, thereby sidestepping the problem of choosing an explicit kernel function.

For Laplacian Eigenmaps, for example, the corresponding kernel matrix becomes the pseudo-inverse of the graph Laplacian. Regarding Isomap, the kernel matrix is $\mathsf{K}_{\text{Iso}} = \mathsf{H}\mathsf{D}_{\text{Iso}}\mathsf{H}$, where D_{Iso} is the matrix of squared approximate geodesic distances. However, as pointed out in Section 3.6, this matrix is not guaranteed to be positive semidefinite, in which case the kernel PCA analogy does not hold. In [Choi and Choi, 2004], a solution

to this problem is suggested, which involves adding constants to K_{Iso} to make it positive semidefinite.

Within the kernel framework it is becomes possible to incorporate test samples, not used in the construction of the reconstructed embedding space, and compute its projection onto the reconstruction [Bengio et al., 2003]; cf. Section 3.4. Naturally, this is useful since we normally do not wish to recompute our reconstructed embedding space as soon as a new sample comes into our hands.

In *Maximum Variance Unfolding* [Weinberger et al., 2004], full use is made of the connection between kernel PCA and the graph-based methods. Using semidefinite programming, a positive semidefinite kernel matrix is learnt from the data such that it has maximum trace (i.e., feature space variance) under the constraint that $|\Psi(x_i) - \Psi(x_j)|^2 = |x_i - x_j|^2$ for neighboring points i, j.

Chapter 4

Approximate geodesic distances reveal biologically relevant structures in microarray data

Jens Nilsson, Thoas Fioretos, Mattias Höglund and Magnus Fontes

Abstract

Motivation: Genome-wide gene expression measurements, as currently determined by the microarray technology, can be mathematically represented as points in a high-dimensional gene expression space. Genes interact with each other in regulatory networks, restricting the cellular gene expression profiles to a certain manifold, or surface, in gene expression space. To obtain knowledge about this manifold, various dimensionality reduction methods and distance metrics are used. For data points distributed on curved manifolds, a sensible distance measure would be the geodesic distance along the manifold. In this work, we examine whether an approximate geodesic distance measure captures biological similarities better than the traditionally used Euclidean distance.

Results: We computed approximate geodesic distances, determined by the Isomap algorithm, for one set of lymphoma and one set of lung cancer microarray samples. Compared to the ordinary Euclidean distance metric, this distance measure produced more instructive, biologically relevant, visualizations when applying multidimensional scaling. This suggests the Isomap algorithm as a promising tool for the interpretation of microarray data. Furthermore, the results demonstrate the benefit and importance of taking nonlinearities in gene expression data into account.

4.1 Introduction

The study of gene expression data has been greatly facilitated by the development of the microarray technology. High density oligonucleotide arrays [Lockhart et al., 1996] and

cDNA microarrays [Schena et al., 1995a,b] measure the expression of thousands of genes simultaneously. Comparing the transcription profiles of different types of tissue specimens permits the identification of genes that best distinguish the samples. When samples correspond to different pathological states of the same tissue, or subtypes of the same malignancy, transcription profiling holds promise as a method for classifying cancers from a molecular rather than from a morphological perspective [Ramaswamy et al., 2001, Pollack et al., 2002, Khan et al., 2002]. Furthermore, complex biological processes, such as the onset of the cell cycle [Iyer et al., 1999] or cellular responses elicited by various growth factors [Fambrough et al., 1999], are now open for a detailed analysis by the study of dense time series.

A main problem in microarray data analysis is how to extract the central features of the vast amount of information generated. Mathematically, the expression profile of a sample can be represented as a point in a gene expression space with coordinates given by its expression levels. Put in another way, the location of a cell sample in gene expression space is determined by its transcriptional state. Genes interact with each other in regulatory networks and as a consequence, the functional relations between genes restrict the distribution of possible gene expression states of the cell to some manifold, or surface, in gene expression space. Typically, the number of genes measured is very large and consequently, so is the dimension of the studied gene expression space. A variety of mathematical methods have been described that reduce the dimensionality of the data sets so as to find the principal features of the data [Quackenbush, 2001]. Two established and commonly used unsupervised methods are Multidimensional Scaling (MDS) and Principal Component Analysis (PCA) (see e.g. [Alter et al., 2000] and [Bittner et al., 2000] for applications to expression data). These methods work best when data are linearly distributed in data space. For the more general case of nonlinearly distributed data, there are several dimensionality reduction methods like e.g. Principal Curves [Hastie and Stuetzle, 1989] or Kernel PCA [Schölkopf et al., 1998], but so far methods like these have been sparsely applied to gene expression data.

A natural way to handle nonlinearities is to adopt a different distance metric in data space. In most of the applied methods, Euclidean metrics or correlation is applied when estimating similarities/differences between biological samples. In the present investigation we have applied geodesic distances as an alternative measure for similarity. As opposed to the straight-line Euclidean distance, geodesic distances are measured along the surface of the manifold on which data is assumed to lie. Approximations of the geodesic distances are calculated using the Isomap algorithm, originally described by Tenenbaum et al. [2000] and developed as a tool for analysis of complex data, such as e.g. digital images. Isomap tries to approximate the data manifold by a graph, constructed by locally connecting nearest neighbors. Approximate geodesic distances are then calculated as the distance of the shortest paths between samples in the graph. In the present study we have applied the approximate geodesic distance measure on two previously analyzed data sets - one set of lymphomas [Alizadeh et al., 2000] and one set of lung cancer tumors [Garber et al., 2001], and shown that this approach reveals biologically relevant structures in the data not easily detected with a standard multidimensional scaling analysis of the same data using Euclidean metrics.

4.2 Systems and Methods

Data Sets. Two previously described microarray data sets were analyzed – one set of 96 lymphoma samples [Alizadeh et al., 2000] and one set of 73 lung cancer samples [Garber et al., 2001]. The selection of genes was, in both cases, unsupervised. The lymphoma data was filtered so that the fluorescent intensity in each channel was greater than 1.4 times the local background for a gene to be included in the analysis, resulting in a total of 854 genes. The samples were divided into the nine diagnostic classes defined by Alizadeh et al. [2000]. The lung cancer data was centered by sample mean and filtered so that the raw intensity in both channels was greater than or equal to 1.5 times the background, resulting in 831 genes. Samples were divided into five diagnostic classes as described by Garber et al. [2001].

Multidimensional Scaling. Multidimensional Scaling (MDS) is a mathematical procedure that creates a lower-dimensional configuration of points $\{\vec{x}_i'\}$ so as to optimally approximate given distances between points $\{\bar{x}_i\}$ in a higher-dimensional space. MDS was performed using an implementation of non-metric MDS [Schiffman et al., 1981] available in the STATISTICA 6.0 software (Statsoft, Tulsa, OH). In short, the algorithm minimizes the raw stress defined as

$$\varphi = \sum_{ij} (d(\vec{x}_i', \vec{x}_j') - f(d(\bar{x}_i, \bar{x}_j)))^2$$

for different functions f belonging to a set M of monotone functions. The effect of the transformations f is such that the order relation between distances is preserved rather than the absolute values. The optimization procedure alternates between minimizing φ over M and the set of lower-dimensional configurations. The initial configuration in the optimization is found through Principal Component Analysis, i.e. by setting f to the identity.

Isomap. Generally, MDS-techniques work with distance data as it is given, possibly letting them undergo some monotone transformation as described above. The Isomap algorithm [Tenenbaum et al., 2000] differs in this respect since distances are transformed so that nonlinear dependencies in data are taken into consideration. Assume, for example, that data are sampled from a spiral-shaped configuration (Figure 4.1). Then the preferable distance measure between points is perhaps not the Euclidean distance, but the geodesic distance along the spiral. Consequently, in Figure 4.1, the distance between a and b should be considered shorter than that between a and c.

To handle this, Isomap constructs a graph G locally by connecting each data point to its nearest neighbors. The set of nearest neighbors of a point $\bar{x}_0$ is defined either as all points $\bar{x}_i$ within a distance $d(\bar{x}_0, \bar{x}_i) < \varepsilon$, for some chosen $\varepsilon > 0$, or as the K closest points, for some chosen integer $K > 0$. After the graph construction, approximations $d_G(\bar{x}_i, \bar{x}_j)$ to the geodesic distances between points $\bar{x}_i, \bar{x}_j$ are calculated by finding the shortest path in the graph between $\bar{x}_i$ and $\bar{x}_j$. MDS is then applied to these approximate geodesic distances instead of the original distances.

The ability of the Isomap algorithm to produce good approximations of the geodesic distances on the underlying manifold depends on the density of data points and the choice

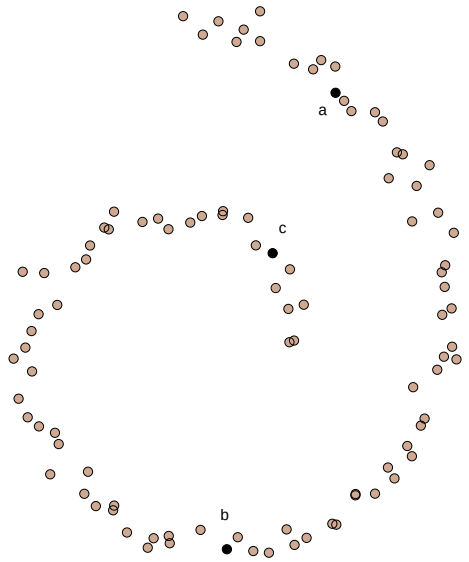

Figure 4.1: Data distributed along a spiral. The geodesic distance along the spiral is presumably more reasonable than the Euclidean distance. Thus the distance between a and b should be considered shorter than that between a and c.

of K (or ε) [Bernstein et al., 2000]. If the parameter is too small, a single connected graph is not achieved and distances can not be calculated between all sample pairs. If, on the other hand, the parameter is too large, shortcuts, not following the surface of the manifold, may appear in the graph. For the K-rule, the latter situation is likely to appear for large parameter values and low data density. It is reasonable to assume that the dimension of the underlying nonlinear data manifold is fairly large, thus the densities of the presently analyzed data sets are expected to be low. With this in mind and after trying different parameter values, we chose to construct the graph using the K-rule with $K = 2$.

Projection Quality. The accuracy of an MDS approximation is quantified by the raw stress of the final point configuration. Lower stress values correspond to a better approximation of the original distances. To evaluate how well an individual sample $\bar{x}_i$ is represented in a projection one can calculate the raw stress over the distances between $\bar{x}_i$ and all other samples, i.e. $\varphi_i = \sum_j (d(\bar{x}'_i, \bar{x}'_j) - f(d(\bar{x}_i, \bar{x}_j)))^2$. Samples with higher stress values are then less well approximated by the projection than samples with lower stress values.

Since the calculation of Isomap graph distances depends on the distribution of data it is desirable to investigate how stable an acquired Isomap visualization is to changes in the data. This can be done by excluding one sample at a time, constructing Isomap graphs for each of the remaining data subsets and noting for which samples the Isomap graph structure changes drastically. Let G_0 be the graph that is constructed when the whole data set is used and let G_i be the resulting graph when the ith sample is left out. For each left-out sample we calculate δ_i, the Euclidean norm of changes in graph distance between

points present in both G_0 and G_i divided by the Euclidean norm of graph distances in G_0 between points present in both G_0 and G_i as

$$\delta_i = \frac{\sqrt{\sum_{k,l \in J \times J} (d_{G_0}(\bar{x}_k, \bar{x}_l) - d_{G_i}(\bar{x}_k, \bar{x}_l))^2}}{\sqrt{\sum_{k,l \in J \times J} d_{G_0}(\bar{x}_k, \bar{x}_l)^2}}$$

where $J = \{k; \bar{x}_k \in G_i\}$. Then δ_i is a measure of how deformed the Isomap graph is.

4.3 Results and Discussion

Analysis of the lymphoma data set. To analyze the lymphoma samples, at first a distance matrix based on Euclidean metrics was produced from the data obtained by Alizadeh et al. [2000]. A lower-dimensional representation of the data was obtained by performing non-metric MDS. Without previous knowledge of subclasses within the sample set no distinct clusters were seen. However, when the classification used by Alizadeh et al. [2000] was applied, it was seen that cases belonging to the same classes were mainly located in the same regions of the projection (Figure 4.2a). In marked contrast, a similar MDS analysis of the calculated Isomap distances already produced distinct structures when projected into two dimensions. When samples were marked according to their classification a clear connection between classification and structure appeared (Figure 4.2b). The two-dimensional Isomap visualization revealed three well-separated groups, all consisting of samples previously known to be of divergent origin. These groups were located at the periphery of the projection; one constituting the chronic lymphocytic leukemia (CLL) samples (yellow), one the activated blood-B samples (light blue), and a third group including resting/activated T cells (red) and transformed cell lines (pink). The other samples were positioned in the center of this structure. One interesting observation, already apparent in the two-dimensional representation, was the misclassification of one of the transformed cell lines (pink in Figure 4.2b). This case, SUDHL-5, was grouped together with the other transformed cell lines by hierarchical cluster analysis [Alizadeh et al., 2000]. In contrast, the Isomap algorithm placed this case at a distance from the transformed cell line class and between the activated blood-B and the diffuse large B-cell lymphoma (DLBCL) samples. Hence, this cell line seems to be more similar to the DLBCL and the activated blood-B class, than the other transformed cell lines. This is perhaps not surprising, given the fact that SUDHL-5 is a cell line established from a DLBCL tumor [Epstein and Kaplan, 1979], whereas at least three of the remaining cell lines are of T-cell origin [Tweeddale et al., 1987, Mehra et al., 2002]. The third dimension revealed even further informative structures that could be linked to previous biological knowledge (Figure 4.2c and d). For example, the central group of the samples in Figure 4.2b showed an extended distribution in the third dimension, revealing two arms extending upwards; one consisting of the follicular lymphoma group (FL, green) and the other of the DLBCL group (blue) interconnected by two cases of germinal center B-cells (GC B-cells; orange). When examining the FL cases (green in Figure 4.2c, d and 4.3b) these could be separated into two groups; one located more closely to the GC B-cells and one more closely to the resting blood-B samples. The proximity of the latter group with the resting blood B-cells (violet) and CLL samples (yellow), could reflect the low proliferation rates of these samples, as also suggested

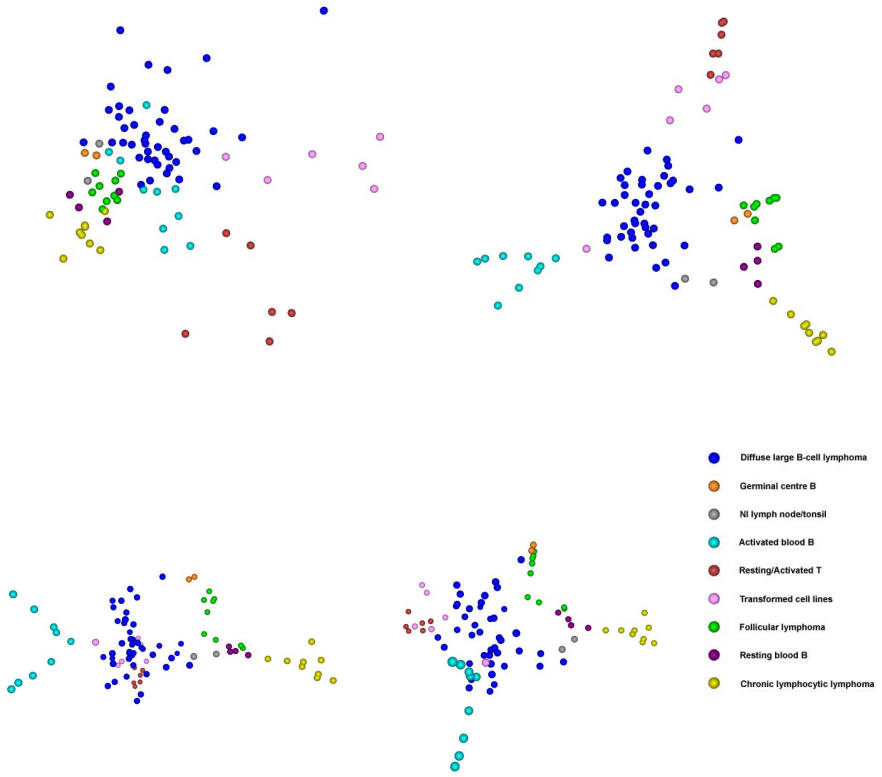

Figure 4.2: Visualization of lymphoma microarray data. a) A two-dimensional MDS representation of the Euclidean distances. b) A two-dimensional MDS representation of the approximate geodesic distances. c) A three-dimensional MDS representation of the approximate geodesic distances. d) As in Figure 4.2c but from a different angle. Color codes in Figure 4.2a-d are as given in the figure.

by Alizadeh et al. [2000]. The largest and most heterogeneous group of tumors was the DLBCL, which formed an extended central cluster (2c and d). When labelling this group into those belonging to the 'germinal center B cell-like' (GCBL) or 'activated B cell-like' (ABL) DLBCL types as described by Alizadeh et al. [2000], the GCBL cases occupied the upper-half of the structure (red in Figure 4.3a), whereas the ABL group preferentially occupied the lower half (green in Figure 4.3a). As expected, the GCBL group extended towards the two GC B-cell samples (orange in Figures 4.2c, d and 4.3b), whereas the ABL group was positioned in-between samples of normal lymph node/tonsil (gray in Figure 4.2 b-d) and the activated blood B samples (light blue in Figure 4.2 b-d). The proximity of the GCBL and ABL tumors to these normal cell samples was also seen by Alizadeh *et al.* using hierarchical cluster analysis [Alizadeh et al., 2000]. Interestingly, a close inspection of the DLBCL cases (blue in Figure 4.3b) extending upwards towards the GC B-cell samples (orange in Figure 4.3b), revealed that these in fact were t(14;18)-positive as recently reported by Huang et al. [2002]. Thus, Isomap placed tumors with similar primary genetic changes, i.e. DLBCL with a t(14;18) and FLs, which are known to be characterized by the same translocation, in close proximity and in a continuum, extending out from the normal GC B-cells. Hence, the data suggest that the latter two tumor types both initially develop from GC B-cells as previously suggested [Alizadeh et al., 2000, Küppers et al., 1999]. In addition, as the tumor samples are organized in a linear order, originating from the GC B-cells, the observed order could possibly reflect gene expression alterations related to tumor progression. Finally, when identifying the individual samples within the activated blood-B samples, it was found that the upper arm (red in Figure 4.3c) corresponded to cells stimulated for more than 24 hours, whereas the lower arm (green in Figure 4.3c) included the samples stimulated for 6 hours. Hence, this observation further underscores the ability of the Isomap algorithm to differentiate between biologically similar samples.

Analysis of the lung cancer data set The same analysis was applied to the lung cancer data set [Garber et al., 2001]. First, a two-dimensional MDS analysis was performed based on Euclidean distances, displaying an unstructured cluster of tumor cases (Figure 4.4a). When the classification used by Garber *et al.* was applied, it became evident that one half of the structure was dominated by adenocarcinoma cases (AC; red) and the other half by squamous cell carcinoma cases (SCC; black). In contrast, the approximate geodesic distances revealed further substructures when performing the corresponding MDS analysis (Figure 4.4b). More specifically, the SCCs were separated further from the ACs. In addition, all but one of the small cell lung cancer cases (SCLC; green) were located within or adjacent to the SCC cluster. The remaining case (207-97-SCLC) was placed together with the ACs, suggesting a larger similarity of this tumor to that group. Further, five out of the six normal cases (blue) formed a well separated group at the periphery. These cases were derived from adult tissue, whereas the outlier, located among the AC tumors, was a sample obtained from fetal lung.

A three-dimensional MDS-projection (Figure 4.4c and d) based on approximate geodesic distances displayed an even better separation between the three major groups - ACs, SCCs and normal cases. Furthermore, the SCCs and the SCLCs, which clustered together in the two-dimensional visualization were now separated. Like in the two-dimensional projection, the ACs formed one heterogeneous group. Thus we could not confirm the results

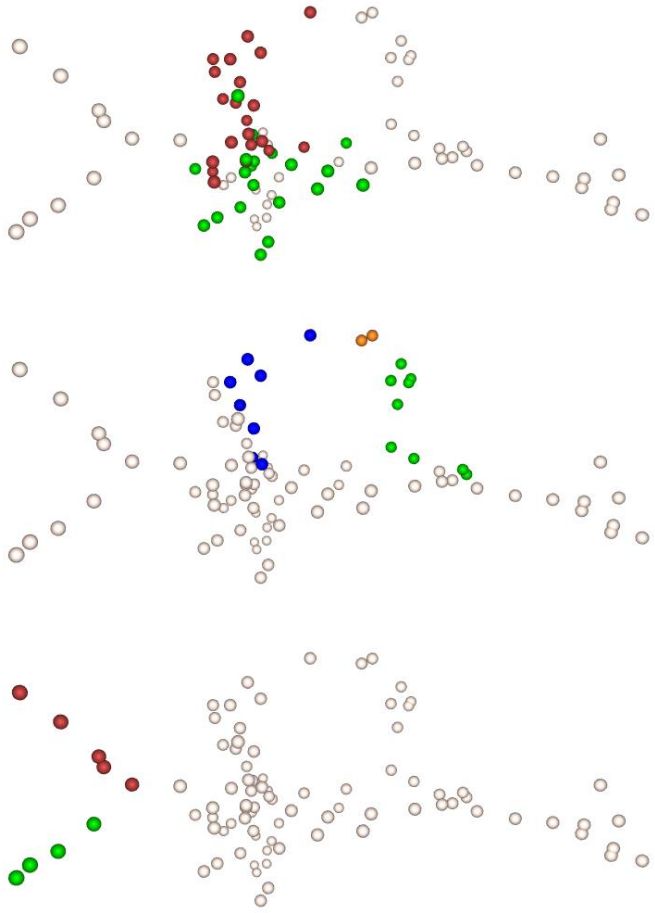

Figure 4.3: Visualization of lymphoma microarray data. a) DLBCL-GCBL, red; DLBCL-ABL, green. b) Follicular lymphomas, green; germinal center B-cells, orange; DLBCLs with t(14;18), blue. c) Blood-B cells activated for 6 hours, green; blood-B cells activated for 24 hours, red. For details, see text.

of Garber *et al.* who, using hierarchical clustering, divided the ACs into three major subgroups and a fourth group of six samples, not included in any of these main clusters. It remains unclear whether this discrepancy stems from a shortcoming in the Isomap algorithm´s capability to identify these suggested subgroups, or from the fact that hierarchical clustering always detects clusters in data regardless of whether any real underlying groups are present. Similarly, the large cell lung cancer cases (LCLC; violet) could not be separated from the ACs. These tumors are poorly differentiated and their expression similarities with ACs may suggest a common tumor origin.

Projection Quality. Additional Isomap projections of the lymphoma data with dimensions from four up to nine were made. For each projection dimensionality, the overall raw stress was calculated and plotted in a scree plot (Figure 4.5a). The scree plot indicated that the data would be well described by a three- or four-dimensional projection. Raw stress values were also calculated for individual samples, in order to evaluate the credibility of sample locations. Two samples, OCI Ly10 and DLCL-0011, had a substantially higher stress than the rest and these were marked in the visualization (Figure 4.5b). However, none of these samples were crucial for the detailed biological interpretations made. In order to evaluate the robustness of the structure, 96 Isomap graphs were constructed, excluding one sample at a time. For each sample subset, the distance deviations in the Isomap graph were calculated (Figure 4.5c). For the studied data set and the used Isomap parameter settings, the samples with high graph distance deviations are apparently important in the calculation of graph distances and noise disturbances on these samples have a relatively large impact on the graph structure. Since there is no knowledge of the underlying data manifold, we can not tell to what degree their positions in gene expression space are 'biologically correct' or if they have been dislocated by noise.

Raw stress analysis was performed also for the lung cancer data. A scree plot showed that a three- or four-dimensional projection was appropriate. To evaluate the goodness of fit for individual samples, individual raw stress values were calculated. The distribution of these values was more homogeneous than the corresponding distribution for the lymphoma data in that it did not contain any obvious outlier values.

4.4 Conclusions

In this work, two alternative ways of measuring dissimilarities or distances between gene expression profiles were compared. Visualizations were created with both Euclidean and approximate geodesic distances as inputs in MDS. The results showed that the approximate geodesic distance measure gave rise to more informative visualizations on the investigated lymphoma and lung cancer data. Even without supervised filtering of the genes with respect to class differentiation, e.g. by creating a weighted gene list [Luo et al., 2001], diagnostic classes appeared as discernible units. That the approximate geodesic distance measure seems more informative could be taken as an indication that tumor samples are distributed on a nonlinear manifold in gene expression space, which in turn would imply that functional relations between genes are nonlinear. Furthermore, the fact that the approximate geodesic distances correspond to the sum of incremental steps between slightly different tumor samples may open the possibility to capture aspects of tumor progression

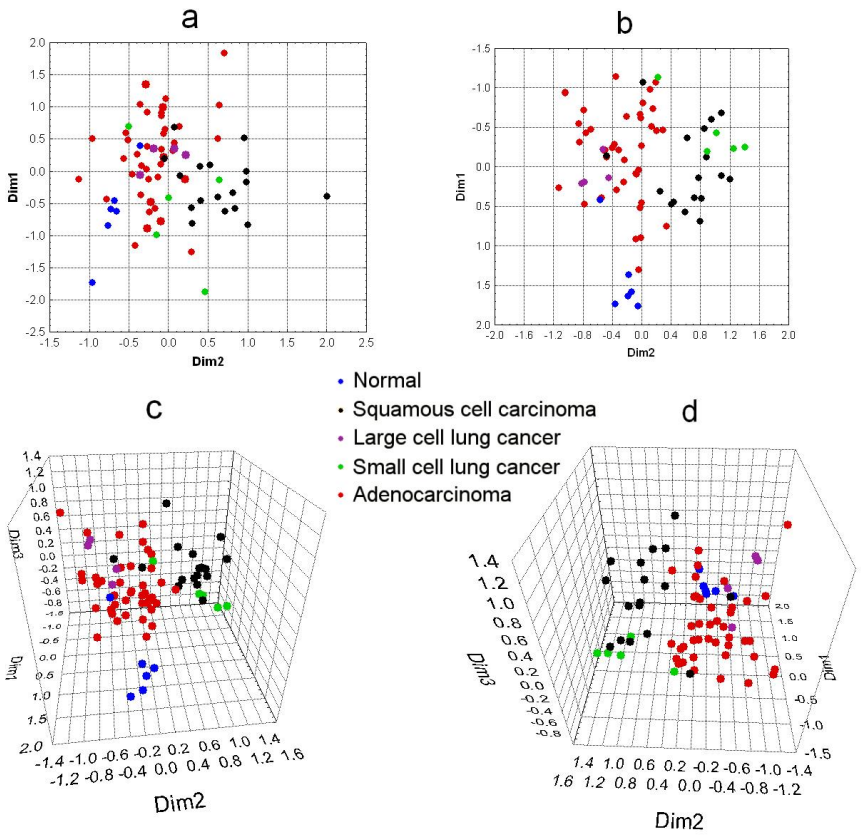

Figure 4.4: Visualization of lung cancer microarray data. a) A two-dimensional MDS representation of the Euclidean distances. b) A two-dimensional MDS representation of the approximate geodesic distances. c) A three-dimensional MDS representation of the approximate geodesic distances. d) As in Figure 4.4c but from a different angle.

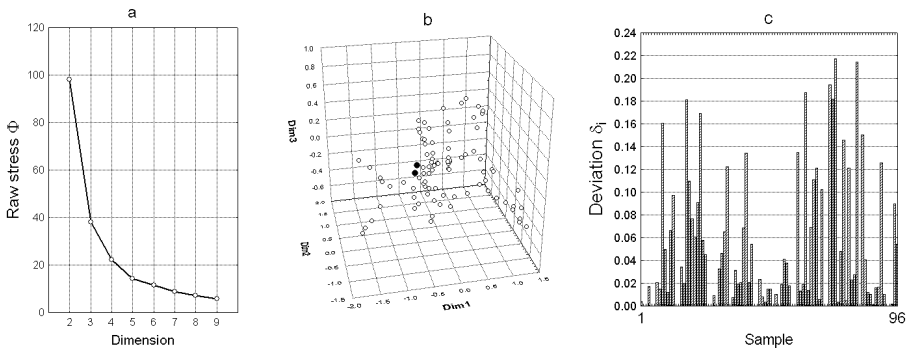

Figure 4.5: Projection quality of lymphoma data. a). Raw stress for MDS-projection of approximate geodesic distances relative to projection dimension. b) Locations of the two samples (OCI Ly10 and DLCL-0011) showing high individual stress values in the three-dimensional MDS representation of the approximate geodesic distances. The representation is shown from the same angle as in Figure 4.2c. c) Structure stability analysis. Deviation in graph distance for each left-out sample.

in the form of microarray data. More generally, the results demonstrate the benefit and importance of taking nonlinearities in gene expression data into account. To conclude, we anticipate that the conceptual framework of geodesic distances will prove useful in both practice and theory for the analysis of gene expression data.

4.5 Acknowledgements

We thank Kalle Åström for stimulating discussions and useful ideas. This work was supported in part by grants from the Swedish Cancer Society.

Chapter 5

Circuit models for manifold learning

Jens Nilsson and Fredrik Andersson

Abstract

Manifold learning and nonlinear dimensionality reduction addresses the problem of detecting, possibly nonlinear, structure in high-dimensional data and constructing lower-dimensional configurations representative of this structure. A popular example is the Isomap algorithm which uses local information to approximate geodesic distances and adopts multidimensional scaling to yield lower-dimensional representations. Isomap is accurate on a global scale in contrast to most competing methods which approximate locally. However, a drawback of the Isomap algorithm is that it is topologically unstable, that is, incorrectly chosen algorithm parameters or perturbations of data may drastically change the resulting configurations. We propose new methods for more robust approximation of the geodesic distances using a viewpoint of electric circuits. In this way, we achieve both the stability of local methods and the global approximation property of global methods, which is demonstrated by a study of the performance of the proposed and competing methods on several data sets.

5.1 Introduction

The field of manifold learning addresses problems in the analysis of data sampled from manifolds embedded in higher dimensional spaces. Such situations arise frequently across diverse disciplines of science such as image analysis, signal processing, psychology and biology. The recent years have brought a growing interest in the development of methods that handle cases where data are sampled from curved manifolds and well-established, linear methods like *Principal Component Analysis* (PCA) [Jolliffe, 1986] and *Multidimensional Scaling* (MDS) [Cox and Cox, 1994] do not perform satisfactory. *Locally Linear Embedding* [Roweis and Saul, 2000], *Laplacian Eigenmaps* [Belkin and Niyogi, 2003] and *Hessian*

Eigenmaps [Donoho and Grimes, 2003] are results of recent efforts to address such situations. Another example in this regard, is the Isomap algorithm [Tenenbaum et al., 2000], which, given Euclidean distances, computes approximations of the geodesic distances on the manifold. Isomap uses a graph-based approach to compute approximate geodesic distances on the data manifold, which are used as input in an MDS algorithm that produces lower dimensional representations of data. One problem that may arise in this process is that of topological instability — small perturbations on data points or minor changes in parameter values may result in large differences in the constructed approximate distances; cf. [Balasubramanian et al., 2002]. In this paper we present algorithms for distance approximation that are more robust against topological instabilities. We study the performance of the methods on several data sets and discuss the influence of different choices of parameter values on the results. We find that our proposed methods are substantially more robust than the Isomap algorithm while retaining good performance on the global scale.

5.2 Manifold learning

Let $\mathcal{X} \subset \mathbb{R}^n$ be a smooth manifold with intrinsic dimension $p \leq n$, and suppose that there exists a coordinate space $\mathcal{Y} \subset \mathbb{R}^p$ and a smooth, bijective mapping $\Phi : \mathcal{Y} \longrightarrow \mathcal{X}$. Assume that we are given data points $\hat{\mathcal{X}}_m = \{x_1, x_2, \ldots, x_m\} \subset \mathcal{X}$, which we refer to as *input coordinates*, to which there correspond *embedding coordinates* $\hat{\mathcal{Y}}_m = \{y_1, \ldots, y_m\}$, satisfying

$$x_j = \Phi(y_j),\ j = 1, \ldots, m.$$

The task of manifold learning is to, given $\hat{\mathcal{X}}_m$, estimate properties of $\mathcal{X}$. Related to this is the dimensionality reduction problem which concerns finding an estimation of the lower-dimensional embedding configuration $\hat{\mathcal{Y}}_m$. The main objective is thus to map $\hat{\mathcal{X}}_m$ onto *reconstructed embedding coordinates* $\hat{\mathcal{Z}}_m = \{z_1, \ldots, z_m\}$, so that $\hat{\mathcal{Z}}_m$ 'represents' $\hat{\mathcal{Y}}_m$ well.

If Φ is a linear isometric embedding, the standard methods of PCA and classical MDS are useful tools for manifold learning. For more general mappings, other techniques are required. In Tenenbaum et al. [2000] the *Isomap* algorithm is introduced, which, under the condition that $\mathcal{Y}$ is convex, can be used to treat isometries Φ. Subsequently, the *C-Isomap* [de Silva and Tenenbaum, 2003] algorithm was suggested as a generalization of Isomap for the broader class of conformal mappings.

Given a metric $d_{\mathbb{R}^n}$ on $\mathbb{R}^n$, Isomap calculates approximations of the geodesic distances $d_{\mathcal{X}}$ on $\mathcal{X}$. An adjacency graph G is constructed by connecting the data points to neighboring points. Two points x_i and x_j are said to be neighbors if either, given an $\varepsilon > 0$, $d_{\mathbb{R}^n}(x_j, x_k) < \varepsilon$, or if, given an integer K, either one of x_j and x_k, has the other one as one of its K closest points. Once the graph is constructed, approximations $d_{\mathrm{ISO}}(x_j, x_k)$ to the geodesic distances $d_{\mathcal{X}}(x_j, x_k)$ on $\mathcal{X}$ are calculated by finding the shortest path in the graph between x_j and x_k. If data points are densely enough sampled from the manifold, d_{ISO} will approximate $d_{\mathcal{X}}$ well. Hence, since Φ is an isometry, d_{ISO} will also approximate the Euclidean distances $d_{\mathcal{Y}}(y_j, y_k)$ in $\mathcal{Y}$. This allows application of standard linear methods to the distances $d_{\mathrm{ISO}}(x_j, x_k)$, and Isomap adopts classical MDS to find lower dimensional representations of data, such that $d_{\mathrm{ISO}}(x_j, x_k)$ are optimally preserved.

The performance of the Isomap algorithm depends on the density of data points, the

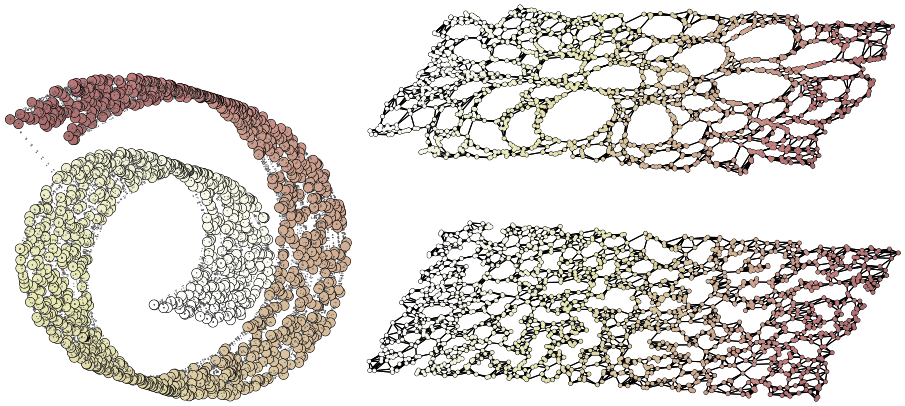

Figure 5.1: To the left is shown 1500 uniformly distributed points ($\hat{\mathcal{X}}_{1500}$) on a swiss roll manifold; the adjacency graph (K=10) contains two shortcuts. To the right, the corresponding 1500 points in $\hat{\mathcal{Y}}_{1500}$, uniformly distributed in the plane (lower) and the corresponding Isomap projection (upper), $\hat{\mathcal{Z}}_{1500}$. The adjacency graph (K=5) is drawn in the configurations. Random fluctuations in data density become amplified in the Isomap reconstruction.

curvature of the manifold $\mathcal{X}$, the amount of noise, and the value of the neighborhood parameter (K or ε); cf. [Bernstein et al., 2000]. For improper parameter choices or in the presence of noise, shortcuts, not following the surface of the manifold, may appear in the graph, disturbing the ability of the algorithm to approximate geodesic distances; cf. Figure 5.1. This has been referred to as the problem of topological instability; cf. [Balasubramanian et al., 2002]. Another property of the Isomap algorithm is that it tends to cluster points in the resulting configurations. When the data point density on the manifold is finite, holes appear in the adjacency graph due to random fluctuations in local density, as illustrated in Figure 5.1 . For pairs of points on opposite sides of such holes the error term $|d_{\mathrm{ISO}} - d_{\mathcal{X}}|$ will become larger and consequently the holes grow in the resulting Isomap projection. Hence, the projection might exhibit structures which are in fact amplifications of random fluctuations in data density. Note that the clustering effect is of local character, in contrast to the error caused by topological instability.

Isomap and C-Isomap are examples of global manifold learning methods, attempting to reconstruct the configuration in $\mathcal{Y}$ correctly on all scales. Local methods, on the other hand, attempt to create lower-dimensional representations that preserve similarities between nearby points but not between faraway points, and are thus more topologically stable. Examples of local methods include *Locally Linear Embedding* [Roweis and Saul, 2000] and *Laplacian Eigenmaps* [Belkin and Niyogi, 2003]. The Laplacian Eigenmaps method, being more topologically stable than Isomap, will serve as a starting point in our development of a more robust distance estimation.

5.3 Circuit models for distance measures

In this section, we present models based on electrical circuits, for the purpose of robust approximation of geodesic distance. Once good such approximations are found, the standard methods of MDS can be used to construct global methods for manifold learning. A circuit interpretation of Laplacian Eigenmaps will serve as starting point, and this model will be modified to enable modelling of charges propagating through the circuit in a moving front fashion.

5.3.1 Charge diffusion in circuits and a circuit interpretation of Laplacian Eigenmaps

Similarly to Isomap, Laplacian Eigenmaps initially constructs an adjacency graph, connecting neighboring points. From the adjacency graph, lower-dimensional representations are found by making an eigenvalue decomposition of the corresponding *graph Laplacian*. The graph Laplacian is a discrete counterpart of the Laplace–Beltrami operator on Riemannian manifolds, and closely related to diffusion and heat flow.

Indeed, it has been noted [Ham et al., 2004] that the Laplacian Eigenmaps method is equivalent to multidimensional scaling of *commute times* under a random walk process on the adjacency graph. Moreover, the commute times can be given an interpretation where the adjacency graph is viewed as an electric network, with each edge represented by a resistor. In this context, the commute time is closely related to the *effective resistance* between nodes; cf. [Doyle and Snell, 2000]. Guided by this observation, we will use the electric circuits in our development of methods for more robust estimation of geodesic distance.

In the approach given in Doyle and Snell [2000] the circuit interpretation of commute times (and thus Laplacian Eigenmaps) consists of a circuit with resistors only, i.e., a counterpart to a standard weighted graph. Here, we will instead use a circuit approach based on diffusion. As a circuit model for graph diffusion, we use circuits where each node has resistive connections to neighboring nodes, and in addition are connected to a common point (ground) by capacitances. Associate each data point x_j with a node n_j in an electrical circuit. To each node, attach a capacitor with capacitance c_j to ground. Based on the metric $d_{\mathbb{R}^n}$, neighboring node pairs $\{n_j, n_p\}$ are connected to each others by resistors r_{jp}, as illustrated in Figure 5.2.

Denote the voltage over capacitor c_j by $v_j(t)$, let $i_{jk}(t)$ denote the current from node n_j to n_k, and let $i_{c_j}(t)$ denote the current from n_j to c_j. According to Kirchhoff's current law,

$$i_{c_j}(t) + \sum_{k=1,k\neq j}^{m} i_{jk}(t) = 0. \tag{5.2}$$

The current between two nodes n_j and n_k is given by

$$i_{jk}(t) = \frac{v_j(t) - v_k(t)}{r_{jk}}, \tag{5.3}$$

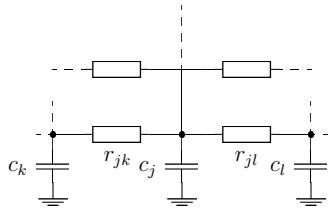

Figure 5.2: The basic RC-circuit

and for the current i_{c_j} the relation

$$c_j \frac{dv_j}{dt} = i_{c_j}(t), \tag{5.4}$$

holds true. By applying (5.3) and (5.4) we can rewrite (5.2) into

$$c_j \frac{dv_j}{dt} = \sum_{k=1,k\neq j}^{m} \frac{v_k(t)}{r_{jk}} - v_j(t) \sum_{k=1,k\neq j}^{m} \frac{1}{r_{jk}}. \tag{5.5}$$

Let $V(t)$ be a column vector containing the elements $v_j(t)$, $j = 1,\ldots,m$. Then (5.5) can be expressed, in matrix form, as

$$C\frac{dV(t)}{dt} = -LV(t), \tag{5.6}$$

where

$$L(j,k) = \begin{cases} \sum_{k\neq j} r_{jk}^{-1}, & \text{if } j = k; \\ -r_{jk}^{-1}, & \text{otherwise.} \end{cases}, \tag{5.7}$$

and C is a diagonal matrix containing the capacities c_j, $j = 1,\ldots m$. The solution to (5.6) is then given by

$$V(t) = e^{-C^{-1}Lt}V_0, \tag{5.8}$$

where V_0 is a column vector containing the initial voltages. The fact that $C^{-1}L$ is real and symmetric (hence diagonalizable) allows the representation

$$e^{-C^{-1}Lt} = \sum_{j=1}^{m} s_j e^{-\lambda_j t} s_j^T, \tag{5.9}$$

where $\lambda_1,\ldots,\lambda_m$ are eigenvalues with corresponding eigenvectors $s_1,\ldots,s_m$. Furthermore, it can be verified that $C^{-1}L$ is positive semi-definite, so $\lambda_j \geq 0$, $j = 1,\ldots,m$.

The p-dimensional reconstruction coordinates obtained by the Laplacian Eigenmaps are in fact given by the p eigenvectors s_i corresponding to the smallest, nonzero, eigenvalues λ_i.

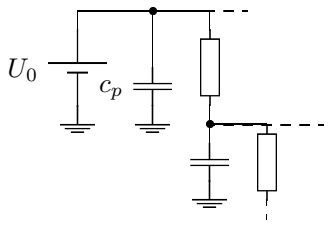

Figure 5.3: The RC circuit with constant voltage source

Let us also briefly remark on the interpretation of Laplacian Eigenmaps in terms of commute times and effective resistance. By effective resistance between two points in a purely resistive circuit, we mean the ratio between an applied voltage between two points, and the induced current. Let us consider (5.6) again. Using the relation (5.4), it can be expressed as

$$I(t) = -LV(t), \tag{5.10}$$

where column vector $I(t)$ contains the currents $i_j(t)$. Hence, the effective resistance is obtained as the pseudo-inverse of $-L$. The pseudo-inverse also appears naturally in the commute time interpretation. From this viewpoint, the elements of the exponential matrix in the right hand side of (5.9) is interpreted as the probability of being in state j starting from state k after time t as pointed out in Ham et al. [2004, p. 6]. Furthermore, it is also noted that the commute time in principle can be obtained by integrating (5.9) (squared, to be specific) over the positive real line. This results in an expression, similar to the right hand side of (5.9) but with $e^{-\lambda_j t}$ replaced with $-1/\lambda_j$, which can be verified to be the pseudo-inverse of $-C^{-1}L$.

5.3.2 The RC and RCZ models

As a charge distribution is applied to the RC circuit depicted in Figure 5.2, currents will move the charges until an equilibrium is reached, at which the voltages over the capacitors are all equal. The dynamic process of charging the capacitors will form the basis in our models for manifold learning, where we use charge times to construct new distance measures.

Instead of the basic model with charges diffusing from an initial state, we adopt a model where one of the nodes, n_p, is attached to a constant voltage source, as illustrated in Figure 5.3. The voltages at the other nodes will then monotonically increase and reach the battery voltage at infinity. By setting the capacity c_p to infinity, we may use equation (5.6) to model the behavior of the circuit. The solution of the system is, analogously to the solution (5.8), then given by

$$V = e^{-C_p L t} V_p, \tag{5.11}$$

where

$$C_p(j,k) = \begin{cases} c_j^{-1}, & \text{if } j = k \neq p; \\ 0, & \text{otherwise;} \end{cases},$$

and

$$V_p(j) = \begin{cases} U_0, & \text{j=p;} \\ 0, & \text{otherwise.} \end{cases}$$

There are several ways to define the distance between points based on the voltages V. One natural choice is to use the commute times discussed above. However, we will instead use the time it takes for a node n_j to reach a certain voltage level $0 < v_{\text{thres}} < U_0$, with a typical value $v_{\text{thres}} = U_0/2$. To assure symmetry we use the mean value of the time it takes to charge node n_j given a constant voltage source at node n_p and vice versa. It should be stressed that these charge times need not fulfil the triangle inequality, why our dissimilarity measure is not necessarily a metric. This, however, does not disqualify the usefulness of the dissimilarity measure. For instance, the well known Mahalanobis measure is not metric; cf. [Chatfield and Collins, 1980].

In what follows, we will refer to the model above as the RC model, and denote the corresponding dissimilarity measure by d_{RC}.

To illustrate how the respective distance measures relate to the geodesic we will consider a set of equidistant samples $\{x_i\}$ from a spiral $\mathcal{X}$ as in Figure 5.4.a. An additional point is positioned between the layers, introducing a shortcut in the adjacency graph for $K \geq 2$. As a first reference we use Isomap, which, disturbed by the graph shortcut, fails to compute distances that relate monotonically to $d_{\mathcal{X}}$ (Figure 5.4.b). A second reference is Laplacian Eigenmaps (Figure 5.4.c) whose embedding distances are clearly more robust with respect to the shortcut but, however, not linearly related to $d_{\mathcal{X}}$. Such nonlinearity is also apparent in d_{RC} which also shows traces of the shortcut influence (Figure 5.4.d). Points close to the source node are charged at a higher rate than points further away and during the time it takes for faraway points to reach threshold potential some (minor in comparison to the closest neighbors, but not in comparison to the ones faraway) charge will have trickled through the shortcut, thus explaining the observed shortcut influence.

This problem stems from the diffusive nature of the charge propagation, and to avoid this issue, we introduce a model in which the nodes are charged through a moving front. In this model we connect nodes directly to the voltage source once the corresponding voltage reaches the threshold level v_{thres}. At this instance, we say that the node reaches *on-state*, and we use the time is takes for the nodes to reach on-state as distance measure.

In this way, nodes neighboring the front are charged directly by their fully charged neighbors, in contrast to being charged indirectly (through points in between) from the original source point. Electronically, we implement the moving front model by replacing the basic RC unit with a slightly more sophisticated one, as illustrated by Figure 5.5. Each node is now equipped with a zener diode and a current controlled switch. As the voltage over the capacitor increases the voltage v_{thres}, the zener diode moves into a conductive phase, turning the switch on, and the node enters the on-state. Hence, we refer to this model as the RCZ model, and denote the corresponding dissimilarity d_{RCZ}.

The purpose of the RCZ model is to make distances more uniformly distributed, since points far from the original source node sooner will reach the on-state. Indeed, applying

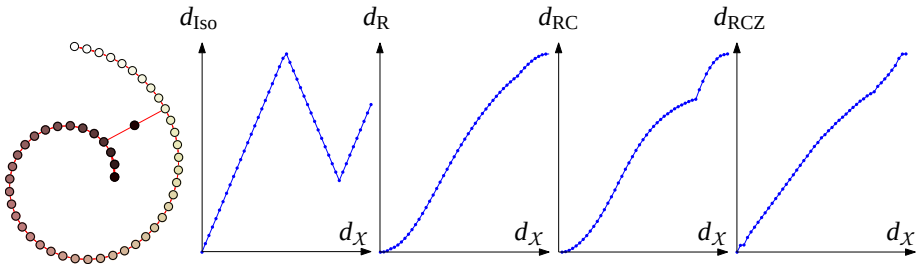

Figure 5.4: The first panel (a) shows perturbed spiral data. The remaining plots show the estimated distances from the upper endpoint to the other points plotted against the true geodesic distances. The distances obtained from Isomap (b) are severely distorted by the presence of the shortcut, while the distances from Laplacian Eigenmaps (c), RC (d) and RCZ (e) deviate less from the geodesic.

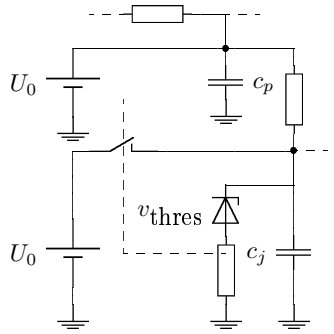

Figure 5.5: RC unit with additional voltage source connected through a zener diode switch.

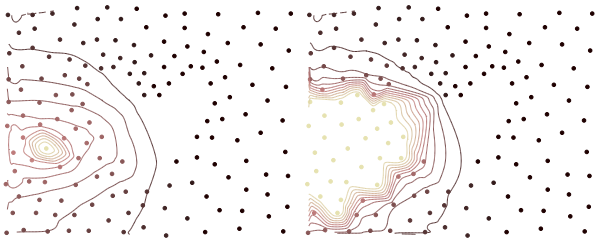

Figure 5.6: Voltage distribution at a certain instant for the RC (left) and the RCZ (right) circuit models

the RCZ model on the spiral data results in distances scaling linearly with $d_{\mathcal{X}}$ showing just a minor influence from the shortcut (Figure 5.4.e).

One physical analogy of the RCZ method is that of combustion propagation. In contrast to ordinary diffusion, the RCZ method relies on a propagating front. At the front, nearby points are charged/heated until they reach a critical value (potential or temperature). In the case of combustion, the material ignites and starts in turn to heat up its environment. In the RCZ model, the ignition role is played by the zener-diod, which causes a sudden increase in the node potential once the threshold voltage is reached. The difference between the RC and RCZ diffusions are illustrated in Figure 5.6. The RCZ diffusion (right panel) clearly illustrates resemblance to a combustion process, while the RC counterpart resemble standard diffusion.

5.3.3 Numerical implementation of RCZ

The natural way to implement the RC and RCZ models is by discretizing the counterparts to (5.6) and stepping forward in time. Note that the RCZ model is identical to the RC model during each time frame of on-state transition. Hence, we simulate RCZ by regarding it as an RC model until a new node reaches critical voltage. The process is then continued as an RC model with new initial values.

Typically, the eigenvalues to the system matrix $(-C_p^{-1}L)$ are of widely varying size. The practical consequence is that nodes close to the initially charged one(s) are charged rapidly, while nodes far away are charged very slowly. Again, this is illustrated in Figure 5.6 (left panel). Hence, much information is obtained at the beginning, which requires a rather small time step h. After some time the dynamics caused by the nodes corresponding to the larger eigenvalues dies out, whereafter the step size seems unmotivatedely small. Systems of ODE's with eigenvalues that that differ greatly in size are commonly referred to as stiff, and usually require special (implicit) treatment. For instance if an explicit scheme such as the explicit Euler,

$$V(t+h) = V(t) - h \cdot C_p L V(t), \tag{5.12}$$

is used, the step size h has to be chosen with respect to the fastest changing node for the scheme to be stable. Hence, even if the fastest changing nodes are not active for a certain solution, the step size has to be chosen with respect to those.

However, the RCZ model does not share the difficulty of stiffness discussed above. Nodes are turned on once the front reaches them, and the charge development shows a uniform appearance in time in contrast with the RC model. Hence, the simple Euler scheme will suffice for modelling the dynamics of the RCZ model. Below, in Algorithm 1, we present an algorithm for the simulation of the RCZ model. By setting the elements of C_p corresponding to the nodes which has reached on-state to zero, we remove their role as variables, and they start to act as sources in the same way as the original source.

Algorithm 1 Computation of RCZ distances

```
for all nodes p
  t=0
  while not all nodes has reached on-state
    t=t+dt
    dV=h C_p A V
    Let V:=V+dV for all nodes that are not in on-state
    Check for nodes that has reached on-state,
      register elapsed time (as distance measure),
      and charge them fully.
  end
end
```

Since only K nearest neighbors are connected in the graph, $C_p^{-1}L$ is sparse. The computational cost for each source node is then $O(Kmt_{\text{tot}})$, where t_{tot} is the total number of time steps. In the RCZ model, nodes are charged at approximately constant speed. It is reasonable to introduce t_{av} as the average time it takes for the front to propagate one node. Then $t_{\text{tot}} \approx mt_{\text{av}}$. In total this gives a time complexity of $O(Kt_{\text{av}}m^3)$. Since the computation of eigenvectors in the MDS step is $O(m^3)$, the complexity above seems asymptotically acceptable.

5.4 Results and discussion

In this section we compare the performance of RCZ, Isomap, Laplacian Eigenmaps and PCA on three different data sets — samples from the Swiss roll manifold, samples from a generated image manifold and finally a set of leukemia gene expression data.

5.4.1 Swiss roll manifold

When both a coordinate space $\mathcal{Y}$ and an input space $\mathcal{X}$ are known (or, specifically, the m-point samples $\hat{\mathcal{Y}}_m$ and $\hat{\mathcal{X}}_m$ are given), we may evaluate the performance of dimensionality reduction methods by comparing the reconstructed m-point configurations $\hat{\mathcal{Z}}_m$ with $\hat{\mathcal{Y}}_m$. A reasonable requirement on any algorithm is that it performs well on affine subspaces. To verify this we apply the method using the embedding coordinates, $\hat{\mathcal{Y}}_m$, as input. Moreover, if $\Phi : \mathcal{Y} \longrightarrow \mathcal{X}$ is isometric, the results should be identical when applying the algorithm to the $\hat{\mathcal{X}}_m$ as when applying it to $\hat{\mathcal{Y}}_m$. Thus, the performance evaluation can be divided with

respect to two properties — the quality of the reconstruction of $\hat{\mathcal{Y}}_m$, and the invariance of the reconstruction under Φ.

When comparing point configurations, only their actual shapes are interesting. Hence, we remove issues of scale, rotation, reflection and translation by fitting the configurations optimally (in a least square sense) to each other using such transformations. Subsequently, the root mean square (RMS) error is taken as a measure of similarity, $E_{\text{RMS}} = \sqrt{\sum_{i=1}^{m} |z_i - y_i|^2 / m}$.

Consider a set of 2000 data points, randomly sampled from a Swiss roll manifold, that is, a rectangle $\mathcal{Y} \subset \mathbb{R}^2$ isometrically transformed into a spiral roll $\mathcal{X} \subset \mathbb{R}^3$ (cf. Figure 5.1). We study the performance of Isomap, Laplacian Eigenmaps, and RCZ applied to this data set. Concerning the parameters, we choose $K = 14$ neighbors, a value where all four methods work well on $\hat{\mathcal{Y}}_m$, and we choose σ-values that work well for the Laplacian Eigenmaps and RCZ methods separately: $\sigma_{\text{LEM}} = \infty$, $\sigma_{\text{RCZ}} = 3\overline{d}$, where $\overline{d}$ is the average edge length in the adjacency graph. Figure 5.7 shows embedding coordinates ($\hat{\mathcal{Y}}_{2000}$) by green/lighter points with the fitted reconstructions ($\hat{\mathcal{Z}}_{2000}$) represented by blue/darker points, and with dotted (red) lines connecting the corresponding points in the configurations. The first row displays reconstructions based on $\hat{\mathcal{Y}}_{2000}$ and the second row shows reconstructions based on $\hat{\mathcal{X}}_{2000}$.

Three main observations can be made: First, Isomap and RCZ reconstructs $\hat{\mathcal{Y}}_m$ globally in a satisfactory way. Laplacian Eigenmaps on the other hand, produces a skewed reconstruction. Second, considering the local error structures in the $\hat{\mathcal{Y}}_m$ reconstructions, a slight clustering effect can be noticed for Isomap, while this effect is stronger using the other two methods. Third, examining the $\hat{\mathcal{X}}_m$ reconstructions, Laplacian Eigenmaps and RCZ reconstructions are reasonably invariant under Φ as required, while the Isomap reconstruction is not. Note that all three methods use the same adjacency graph, so the shortcut is indeed present in all methods.

The lower row of Figure 5.7 shows scatter plots of estimated against true geodesic distances for the respective methods. These results confirm that RCZ is the method that most faithfully estimates geodesic distances in the presence of graph shortcuts.

Obviously, the results depend on the parameter values. For example, with K=10 nearest neighbors, no shortcuts appear in the adjacency graph on the Swiss roll, and Isomap correctly reconstructs the rectangular configuration. In order to thoroughly investigate the method performances over a range of parameter values, we apply the algorithms using various $\sigma \in [\overline{d}, 4\overline{d}]$ and $K = 7, \ldots, 25$. Figure 5.8.a displays the approximation error for RCZ applied to the Swiss roll (transparent surface), and the coordinate space rectangle (wire-frame mesh). Further, the difference between these two error matrices is shown by the lower surface plot, giving an idea of the fraction of error stemming from the geometrical change. The coordinate space error has a local minima around $K = 14, \sigma = 3\overline{d}$. The difference between the two errors behaves more or less monotonic with increased K and σ. Figure 5.8.b shows the error of the Isomap reconstruction over different K for the $\hat{\mathcal{X}}_m$ (rectangle) and the $\hat{\mathcal{Y}}_m$ (Swiss roll), respectively. A sharp increase in error for the Swiss roll appears at K=12, where shortcuts first appear in the graph.

The error for the rectangle configuration is highest at low K and low σ, where the local clustering effect is strongest — a low K gives a higher probability of holes in the adjacency graph, while a low σ gives a stronger punishment of long distances. At intermediate

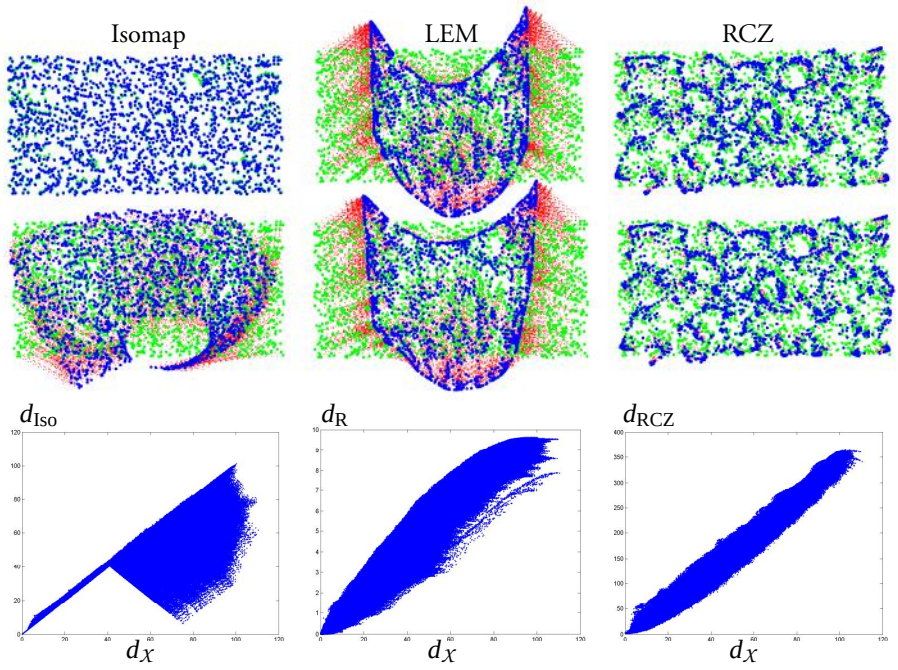

Figure 5.7: Swiss roll reconstructions using Isomap, Laplacian Eigenmaps (LEM) and RCZ; The first and second row shows reconstructions of the embedding and input coordinates respectively. The third row shows scatter plots of estimated against true geodesic distances for the methods applied to the input coordinates.

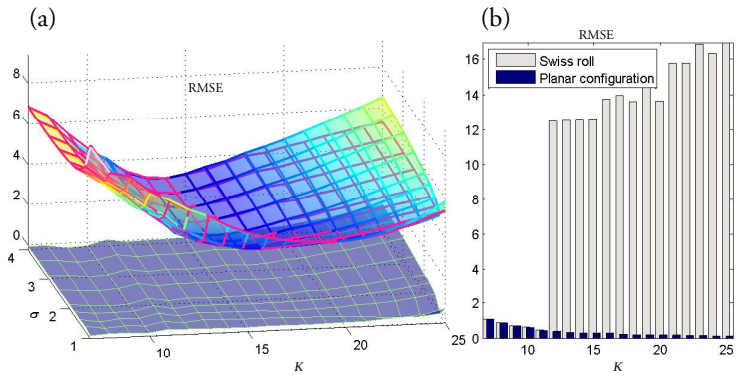

Figure 5.8: (a)RCZ: RMS errors for the planar and Swiss roll configurations, illustrated by the colored surface and red wire-frame, respectively, with their difference, displayed by the lower surface; (b) Isomap: RMS errors for the planar and Swiss roll configurations

parameter values, the error is low, while it increases at higher K and σ. This slightly surprising behavior might be partly explained as a boundary effect — at the boundaries the density of graph edges will be higher, causing a tendency of constricting points along the rectangle edge. This effect grows with K and σ. The fact that the residual error between the rectangle and the Swiss roll grows with K and σ is explained by the increasing risk of shortcuts at larger K and the decreasing capability of down-weighting them at larger σ.

The results from the Swiss roll data set illustrate that the Isomap algorithm is the most accurate, both on local and global scales — *when* it works, that is. Due to topological instability, it is less robust than the other methods. Being a local method, Laplacian Eigenmaps is more stable than Isomap but fails to control the global correctness. Furthermore, it suffers from a larger local clustering error — a problem shared with the RCZ method. Because of its similarity with Laplacian Eigenmaps, we may view the RCZ method as being akin to Laplacian Eigenmaps with global control added. Compared to Isomap we may regard the RCZ method as more robust relatives who pay for the increased robustness with a larger local clustering error.

5.4.2 Image data

In this section, we study the application of PCA, Isomap, Laplacian Eigenmaps and RCZ to a set of 110×80 pixel images picturing a three-dimensional scene with a snowman rotated in different random angles around its axis, illuminated by a light source of random intensity.[1] Having only these degrees of freedom, the underlying dimensionality of the data set is two.

Figures 5.9 and 5.10 display the resulting projections. PCA captures the light intensity in the first component, but rotation angle is not well represented. Conversely, the Laplacian Eigenmaps projection manages to capture the rotational degree of freedom but not

[1] The 3D-scenes were generated using the POV-Ray software and the snowman model is courtesy of Kurt Bangert.

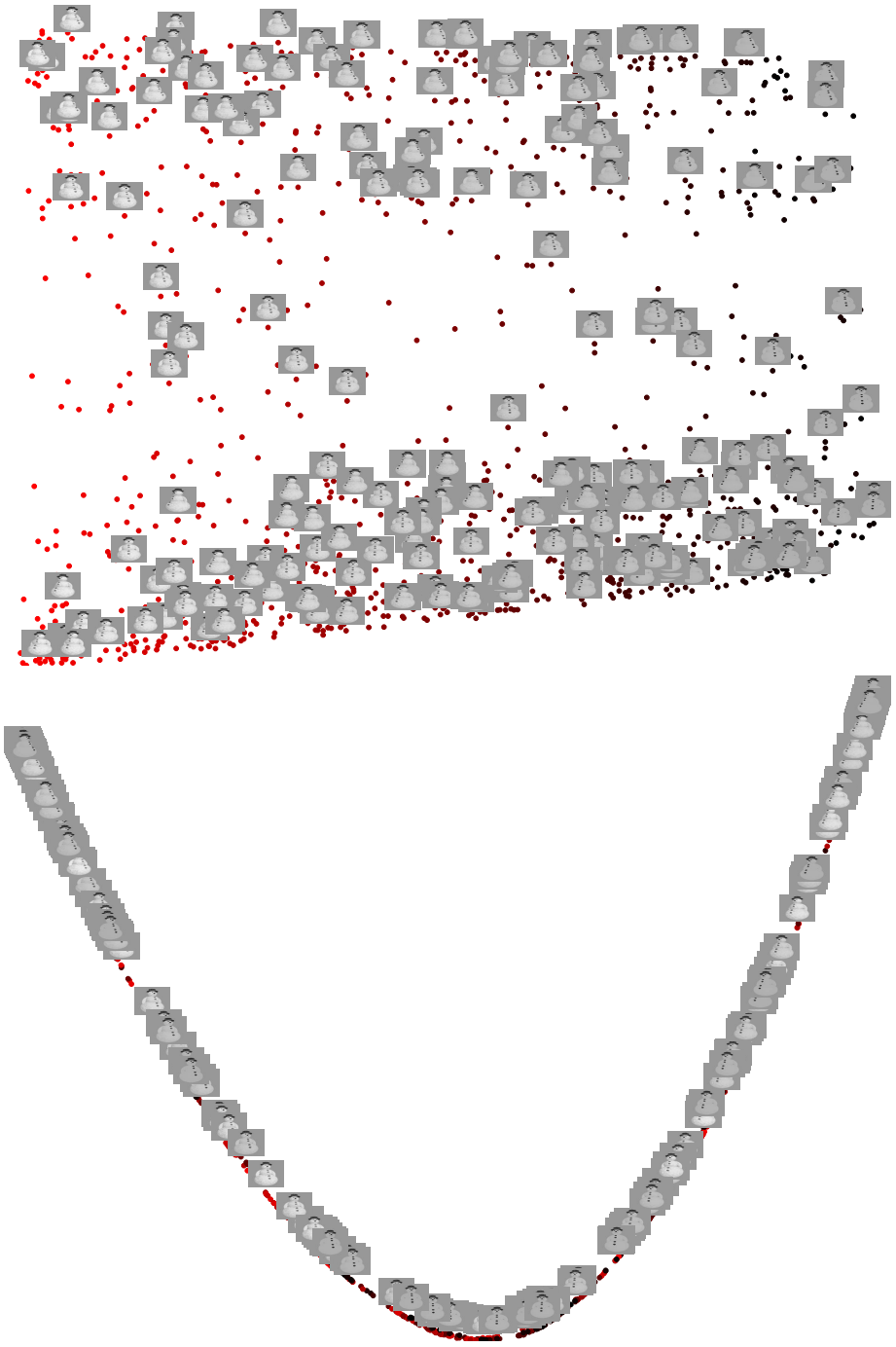

Figure 5.9: Two-dimensional representations of snowman image set using PCA (upper) and Laplacian Eigenmaps (lower). Parameters are set to $K = 10$ and $\sigma = 4\overline{d}$.

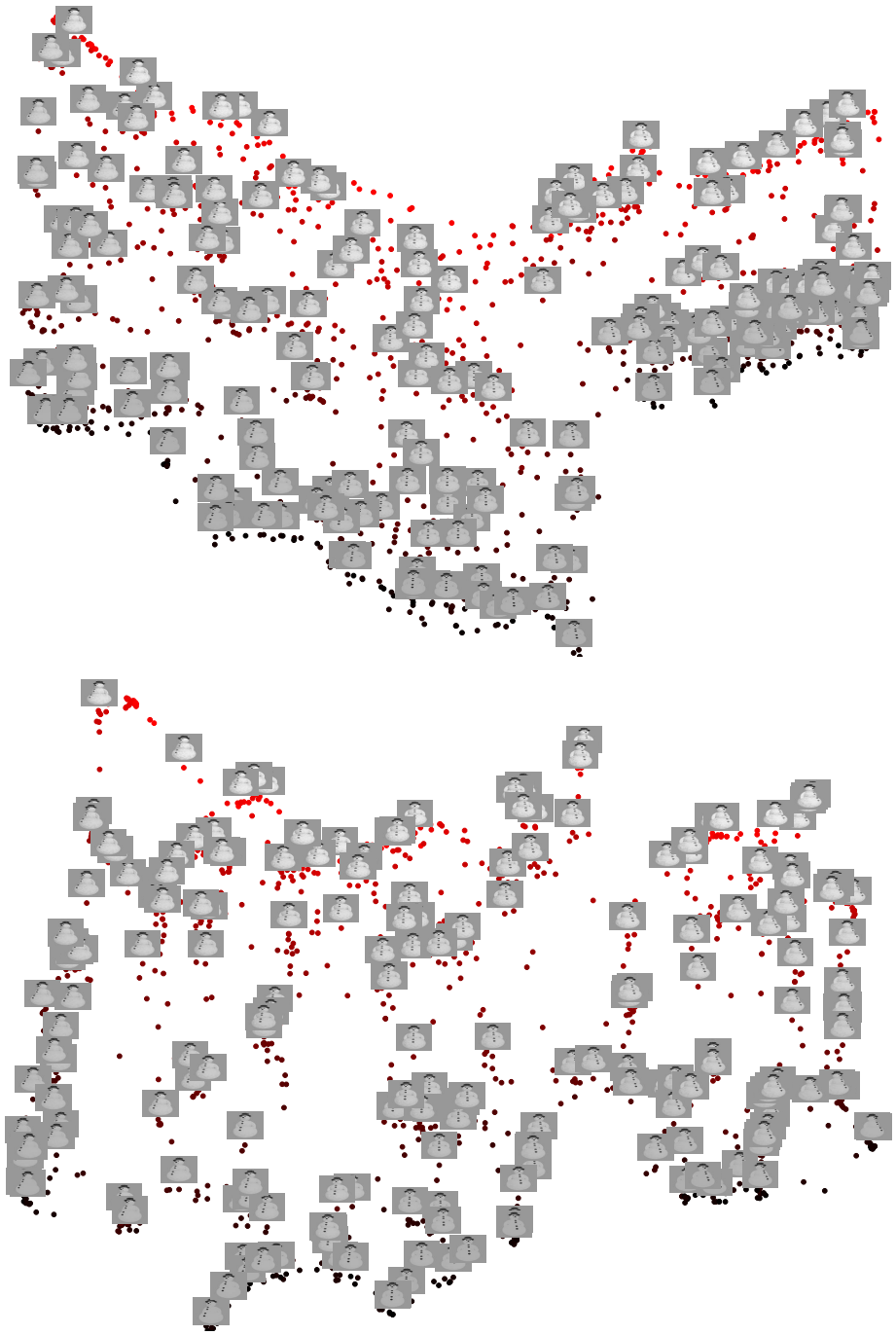

Figure 5.10: Two-dimensional representations of snowman image set using Isomap (upper) and RCZ (lower).

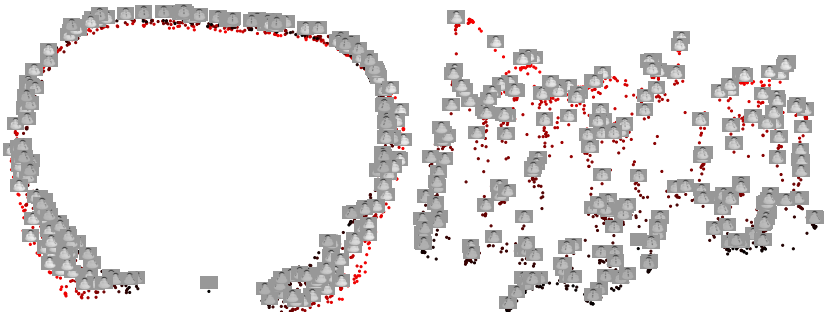

Figure 5.11: Two-dimensional representations of snowman image set with background image constructed using (a) Isomap, and (b) RCZ.

the light intensity, which seems randomly ordered along the one-dimensional structure. Isomap succeeds quite well in extracting the two degrees of freedom. However, the dark right rotated images are somewhat squeezed together. This effect is still present in RCZ, but less strong. Despite some over-clustering, the light intensity and rotation angle are well represented by the projection. The two panels of Figure 5.11 shows the corresponding results from Isomap and RCZ when an empty image with only background has been added to the data set, thus creating a shortcut in the adjacency graph. Again, the results demonstrate the relative topological stability of RCZ compared to Isomap.

For a fixed rotation angle, the variation of light intensity basically takes place along a straight line[2] and therefore it is not surprising that this is the underlying variable that PCA, being a linear method, finds. A three-dimensional PCA projection, capturing 57% of the variation, displays the images as distributed on a segment of a triangular cylinder; cf. Figure 5.12 a. This explains why the two-dimensional PCA projection mixes up rotation angles at some places. Figure 5.12 b shows the parameter space of the image set with the adjacency graph for $K = 10$ overlaid. A typical neighborhood is depicted as a black subgraph, and it is apparent that the neighborhood covers a much wider relative interval in terms of light intensity than rotation. Our suggestion is that this difference in intrinsic scale of the underlying variables makes Laplacian Eigenmaps fail to take the less variant factor into account. In fact, going back to the Swiss roll data and making the parameter plane twice as wide yields similar a Laplacian Eigenmaps behavior — only the most variant factor is captured.

5.4.3 Gene expression data

Dimensionality reduction is frequently used in gene expression data analysis in order to visualize gene expression profiles of cell samples under varying biological conditions [Alter et al., 2000, Nilsson et al., 2004]. The aim is to reveal underlying disease factors and/or discriminate between known and hypothetical subgroups of the samples. As discussed

[2]Neglecting heterogeneous reflection properties, increasing the light intensity is equivalent to multiplying the pixels corresponding to the object by some factor.

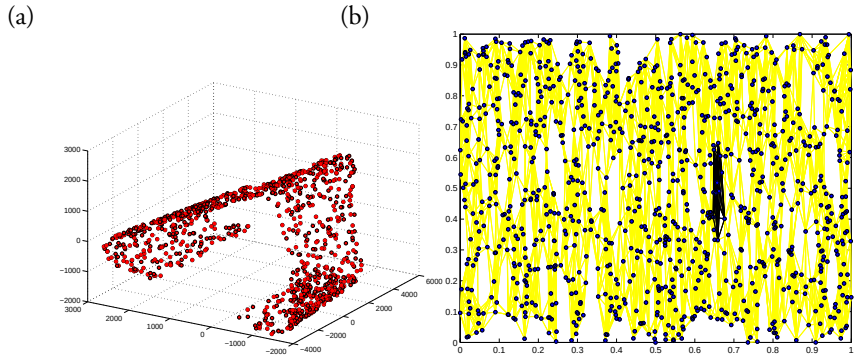

Figure 5.12: (a) Three-dimensional PCA reconstruction of images, and (b) normalized image parameter space with adjacency graph inferred from $\hat{\mathcal{X}}_m$ overlaid. One neighborhood is shown as a black subgraph.

throughout this text, different methods capture different aspects of the underlying structures in data. A way to evaluate the performance of a particular method is to take a data set where samples have been labelled according to a priori clinical knowledge and see whether class structures, such as clusters, are clearly detectable in the reconstruction.

In this section we evaluate the RCZ method for visualization of a set of microarray measurements obtained from various leukemia cancer samples [Ross et al., 2003]. The data consists of expression levels of 2089 genes over 118 samples, divided into six clinically different groups.[3] Figure 5.13 displays two-dimensional representations of the tumor samples created using PCA, Isomap, Laplacian Eigenmaps and RCZ. The parameters were set to $\sigma_{\text{LEM}} = \infty$, $\sigma_{\text{RCZ}} = 3\overline{d}$ and $K = 3$. A brief analysis concludes that Laplacian Eigenmaps and RCZ perform well, and notably better than Isomap, with respect to the clustering of known diagnostic classes. However, not all diagnostic aspects are captured. The separation between the yellow and the dark blue groups appears only in PCA, demonstrating that no single method alone fully separates the groups and that different methods should be used as complements to fully explore different aspects of the data.

5.5 Conclusions

This work demonstrates how an electrical circuit framework enables robust approximation of geodesic distances on an underlying manifold. We use these distances for the purpose of dimensionality reduction, thus constructing nonlinear dimensionality reduction more topologically stable than Isomap. We demonstrate performance on typical data sets and discuss the relations between choice of parameter values and performance for the proposed and competing manifold learning methods, specifically under conditions of sparse or noisy data. Briefly, we find that, relative to the other methods, 1) Isomap is globally and locally

[3] One of the groups (T-ALL) was excluded from the analysis as its gene expressions were very different from the other sample groups.

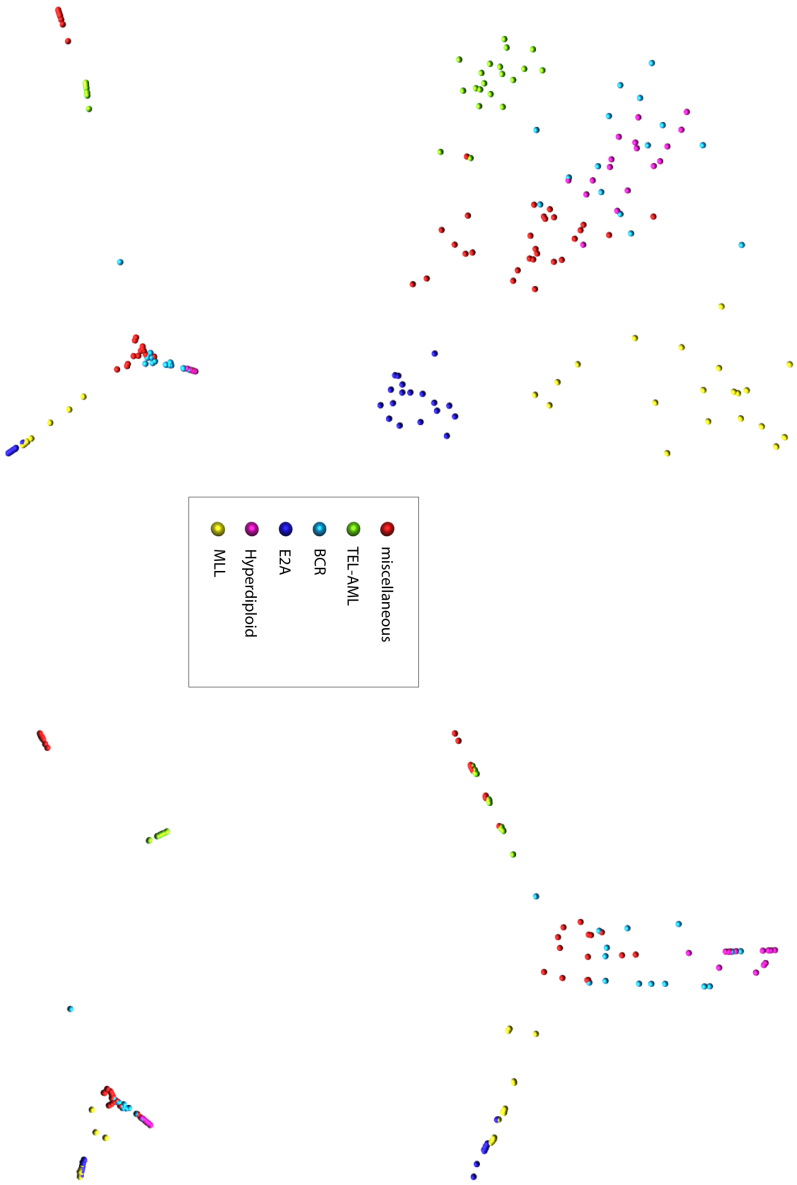

Figure 5.13: Two-dimensional representations of leukemia tumor samples.

well-performing but topologically unstable; 2) Laplacian Eigenmaps is topologically stable but globally and locally less correct; and 3) the proposed method, RCZ, is topologically stable and globally correct but produces larger local errors. The experiments show that the proposed method may be seen as a 'robustization' of Isomap or a 'globalization' of Laplacian Eigenmaps.

Chapter 6

Discussion and outlook

Recall from Section 2.4 the assumption that data are samples from a biologically relevant Riemannian manifold in expression space. This suggests that the use of nonlinear dimensionality reduction should be appropriate. Indeed, in Chapters 4 and 5 results are presented that show that nonlinear dimensionality reduction methods, such as Isomap and RCZ often perform better than linear methods on gene expression data sets. These results demonstrate that the assumption above is useful, even if they do not verify that it is correct. A main reason that stronger conclusions can not be drawn lies in the sparseness of the data. The number of samples m is in the order of 100, while the extrinsic data dimensionality is in the order of 1000 or more. The intrinsic dimensionality of the observation manifold may be much less, but still, most likely, large enough to make the manifold sparsely sampled as $m \sim 100$. In the light of this, can we assume that approximate geodesic distances, as computed by Isomap or RCZ, are close approximations of real geodesic distances? Probably not. On the other hand, it might not be necessary to accurately estimate these in order to produce biologically relevant reconstructions. Instead, it suffices to recover the *ordering* of the samples along the manifold. This explains how, despite the sparseness of data, good results can still be be obtained if data lie on a biologically relevant manifold. Moreover, it motivates the use of non-metric multidimensional scaling in the dimensionality reduction step of Isomap and RCZ.

A related question is whether the problem of topological instability and graph shortcuts is relevant in sparse gene expression data. Clearly, as data becomes sparser it becomes more difficult determine what is a shortcut. In some sense, more or less all graph edges are shortcuts, connecting geodesically distant points on the manifold. Again, what we may hope for is that the approximate geodesic distances are monotonically related to the true geodesic distances. Intuitively, an estimation algorithm like the RCZ increases the chances for this. It is clear, however, that in order to answer this type of questions, theoretical investigations are needed, addressing the question of how dense the data has to be, in relation to the manifold curvature, for the distance approximations to relate monotonically to the true distances.

Figure 6.1 displays the results of Isomap and RCZ on the lymphoma and lung cancer data sets used in Chapter 4. The fact that there are no significant differences in global structure between the Isomap and RCZ reconstructions indicates that, either there were no

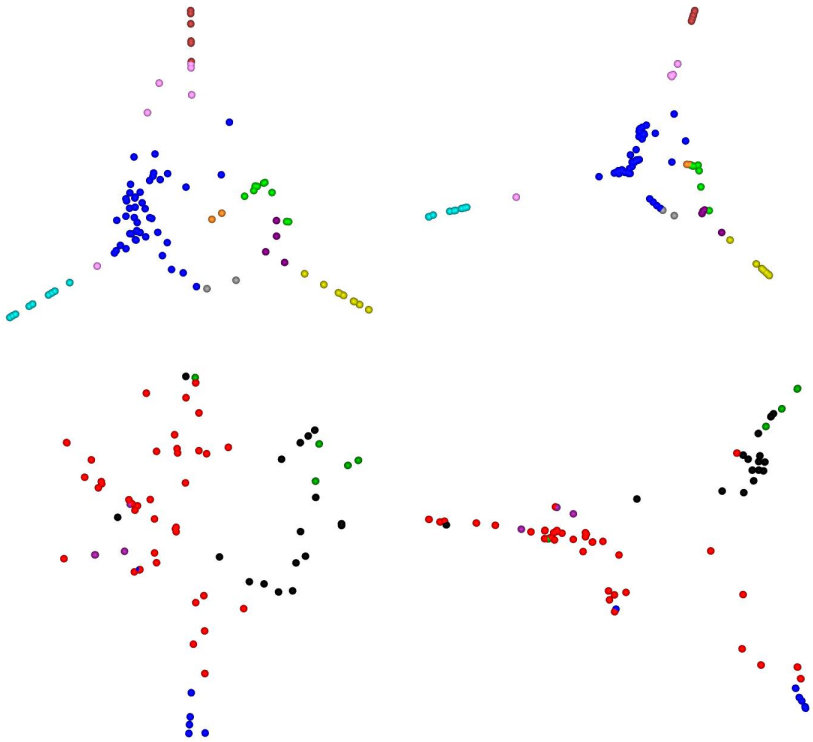

Figure 6.1: Two-dimensional representation of lymphoma data (first row) and lung cancer data (second row) created using classical MDS on Isomap distances (left column) and RCZ distances (right column).

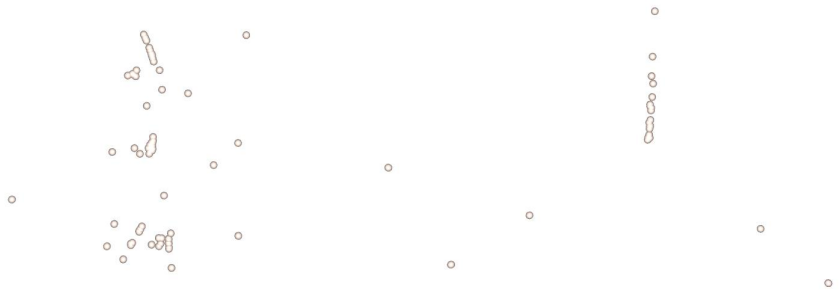

Figure 6.2: Two-dimensional representations of randomly permuted microarray data cancer created using Isomap (left) and RCZ (right).

shortcuts in the data, or, there were, and RCZ could not efficiently handle them. This can also be confirmed by checking that the respective distances in large relate monotonically to each other. Nevertheless, there are other features of RCZ that are useful. In particular the RCZ representation of the lymphoma samples shows more detailed structure than the corresponding Isomap projection. For example, the DLBCL group appears to segregate into three main groups. In Chapter 5 it was pointed out that RCZ, and to some extent Isomap, has a tendency of over-clustering, that is, fluctuations in data density are amplified in the resulting reconstructions. This is a possible explanation of the ability of RCZ to detect finer structure. If there are tendencies of cluster structures in the data, RCZ will discover this more successfully. On the other hand, it is not able to determine whether this structure has a real biological meaning or if it corresponds to spurious random fluctuations. In order to estimate the impact of the over-clustering effect we may apply RCZ and Isomap to data where the expression profile of each sample has been randomly permuted. Figure 6.2 shows the Isomap and RCZ reconstructions of such a random configuration based on the lymphoma data. These representations show some, but little, structure, which indicates that, although present, the over-clustering problem is not extremely severe.

It is worth emphasizing that manifold learning is not only useful for dimensionality reduction. For example, approximate geodesic distances may be used as inputs to clustering algorithms. Results of hierarchical clustering and K-medoids clustering applied to Euclidean, Isomap- and RCZ distances for the lymphoma data are shown in Figure 6.3 and Figure 6.4, respectively. The K-medoids clustering algorithm [Kaufman and Rousseeuw, 1990] is similar to K-means clustering (cf. Section 2.3.2), with the principal difference that only actual data points are used as cluster centers. Not surprisingly, the RCZ distance seems well suited for clustering purposes.

Two parallel themes run through this thesis — gene expression data analysis and nonlinear dimensionality reduction. Both fields are young and still under intense development. To conclude this thesis, I will attempt to point out some directions that seem important to enable full utility of manifold learning and nonlinear dimensionality reduction in gene expression data analysis.

First, the merging of kernel methods and graph-based methods of manifold learning is

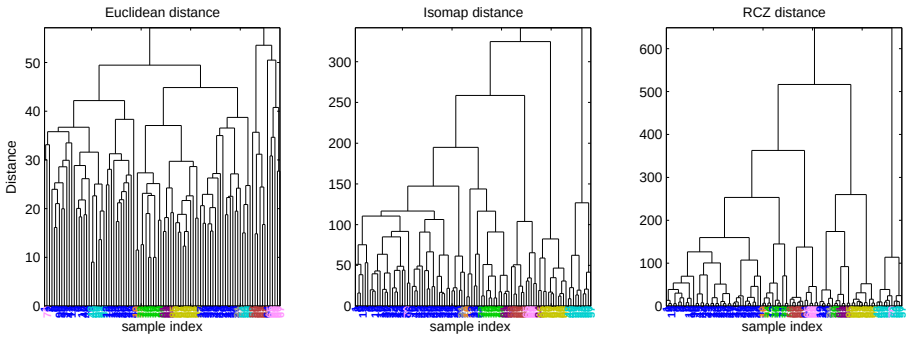

Figure 6.3: Complete linkage hierarchical clustering of the lymphoma data based on Euclidean (left), Isomap (middle) and RCZ (right) distances. The leafs are colored according to diagnostic class (cf. Figure 4.2).

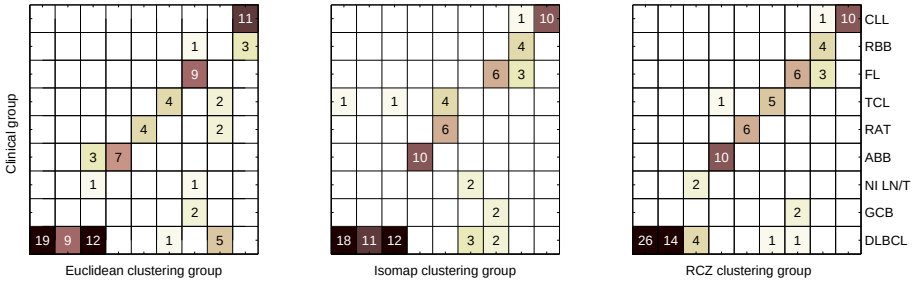

Figure 6.4: Contingency tables for K-medoids (K=9 clusters) clustering of the lymphoma data based on Euclidean (left), Isomap (middle) and RCZ (right) distances. The numbers in the matrix cells i, j denote number of samples of group i in cluster j.

an important development. The kernel field is relatively well-developed and many general problems relevant to graph-based methods have already been studied. An example in this respect is the use of the kernel framework to extend Isomap, Laplacian Eigenmaps and LLE to project test samples [Bengio et al., 2003, Choi and Choi, 2004]. One potentially complicating circumstance here is the fact that no explicit kernel function is available, since the kernel matrix is typically learnt from the data.

Another line of direction, where there seems to be a lot to gain, is the development of alternative ways to construct adjacency graphs, in order to more efficiently capture aspects of the underlying manifold. A lot of efforts have been directed at developing algorithms to extract geometrical properties from the adjacency graph, but most methods adopt the same K-nearest neighbors or ε-graph as the discrete approximation of the underlying manifold. Interesting exceptions in this respect can be found in [Costa and Hero, 2004] and [Carreira-Perpinan and Zemel, 2005].

Concerning the development in biotechnology, it will be of benefit to increase the sampling capacity; both by performing larger single microarray studies and by systemizing the comparison of data sets across studies, but also by incorporating protein expression data in a coordinated way.

Bibliography

A.A. Alizadeh, M.B. Eisen, E.R. Davis, C. Ma, I.S. Lossos, Rosenwald A., J.C. Boldrick, H. Sabet, T. Tran, X. Yu, J.I. Powell, L. Yang, G.E. Marti, T. Moore, J. Hudson, L. Lu, D.B. Lewis, R. Tibshirani, G. Sherlock, W.C. Chan, T.C. Greiner, D.D. Weisenburger, J.O. Armitage, R Warnke, R. Levy, W. Wilson, M.R. Grever, J.C. Byrd, D. Botstein, P.O. Brown, and L.M. Staudt. Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature*, 403:503–511, 2000.

O. Alter, P.O. Brown, and D. Botstein. Singular value decomposition for genome-wide expression data processing and modeling. *Proc. Natl. Acad. Sci. USA*, 97:10101–10106, 2000.

A. Andersson, P. Edén, D. Lindgren, J. Nilsson, C. Lassen, J. Heldrup, M. Fontes, Å. Borg, F. Mitelman, B. Johansson, M. Höglund, and T. Fioretos. Gene expression profiling of leukemic cell lines reveals conserved molecular signatures among subtypes with specific genetic aberrations. *Leukemia*, 19:1042–1050, 2005a.

A. Andersson, T. Olofsson, D. Lindgren, B. Nilsson, C. Ritz, P. Eden, C. Lassen, J. Råde, M. Fontes, H. Morse, J. Heldrup, M. Behrendtz, F. Mitelman, M. Höglund, B. Johansson, and T. Fioretos. Molecular signatures in childhood acute leukemia and their correlations to expression patterns in normal hematopoietic subpopulations. *Proc. Natl. Acad. Sci. USA*, 102(52):19069–19074, 2005b.

A. Arkin, J. Ross, and H.H. McAdams. Stochastic Kinetic Analysis of Developmental Pathway Bifurcation in Phage lambda-Infected Escherichia coli Cells. *Genetics*, 149(4): 1633–1648, 1998.

M. Ashburner, C.A. Ball, J.A. Blake, D. Botstein, H. Butler, J.M. Cherry, A.P. Davis, K. Dolinski, S.S. Dwight, J.T. Eppig, M.A. Harris, D.P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J.C. Matese, J.E. Richardson, M. Ringwald, G.M. Rubin, and G. Sherlock. Gene ontology: tool for the unification of biology. *Nat. Genet.*, 25(1): 25–29, 2000.

G. Bakir, J. Weston, and B. Schölkopf. Learning to find pre-images. In *Advances in Neural Information Processing Systems 16*, 2004.

M. Balasubramanian, E.L. Schwartz, J.B. Tenenbaum, V. de Silva, and J.C. Langford. The Isomap algorithm and topological stability. *Science*, 295(5552):7a–, 2002.

M. Belkin. *Problems of learning on manifolds*. PhD thesis, University of Chicago, 2003.

M. Belkin and P. Niyogi. Laplacian Eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.

Y. Bengio, J.-F. Paiement, and P. Vincent. Out-of-sample extensions for LLE, Isomap, MDS, Eigenmaps, and spectral clustering. Technical Report 1238, Département d'Informatique et Recherche Opérationnelle, Université de Montréal, 2003.

Y. Benjamini and Y. Hochberg. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society B*, 57: 289–300, 1995.

M. Bernstein, V. de Silva, J.C. Langford, and J.B. Tenenbaum. Graph approximations to geodesics on embedded manifolds. Technical report, Stanford University (available at http://isomap.stanford.edu), December 2000.

C.M. Bishop, M. Svensen, and C.K.I. Williams. GTM: The generative topographic mapping. *Neural Computation*, 10(1):215–234, 1998.

M. Bittner, P. Meltzer, Y. Chen, Y. Jiang, M. Seftor, E. Hendrix, M. Radmacher, R. Simon, Z. Yakhini, A. Ben-Dor, N. Sampas, E. Dougherty, E. Wang, F. Marincola, C. Gooden, J. Lueders, A. Glatfelter, P. Pollock, J. Carpten, E. Gillanders, D. Leja, K. Dietrich, C. Beaudry, M. Berens, D. Alberts, V. Sondak, N. Hayward, and J. Trent. Molecular classification of cutaneous malignant melanoma by gene expression profiling. *Nature*, 406:536–540, 2000.

C. Blaschke, J.C. Oliveros, and A. Valencia. Mining functional information associated with expression arrays. *Funct Integr Genomics*, 1:256–268, 2001.

B.M. Bolstad, R.A. Irizarry, M. Åstrand, and T.P. Speed. A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics*, 19:185–193, 2003.

B.E. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. In *Computational Learing Theory*, pages 144–152, 1992.

M. Brand. Charting a manifold. In *Advances in Neural Information Processing Systems 15*, 2003.

P. Broberg. A comparative review of estimates of the proportion unchanged genes and the false discovery rate. *BMC Bioinformatics*, 6, 2005.

M.P.S. Brown, W.N. Grundy, D. Lin, N. Cristianini, C.W. Sugnet, T.S. Furey, M. Ares Jr, and D. Haussler. Knowledge-based analysis of microarray gene expression data by using support vector machines. *Proc. Natl. Acad. Sci. USA*, 2000.

A. Brun, H.-J. Park, H. Knutsson, and C.-F. Westin. Coloring of dt-mri fiber traces using laplacian eigenmaps. In *Proceedings of the Ninth International Conference on Computer Aided Systems Theory*, 2003.

C.J.C. Burges. Geometric methods for feature extraction and dimensional reduction. In *Data Mining and Knowledge Discovery Handbook: A Complete Guide for Practitioners and Researchers*. Kluwer Academic Publishers, 2005.

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

M.A. Carreira-Perpinan and R.S. Zemel. Proximity graphs for clustering and manifold learning. In Lawrence K. Saul, Yair Weiss, and Léon Bottou, editors, *Advances in Neural Information Processing Systems 17*, Cambridge, MA, 2005. MIT Press.

C.R Chatfield and A.R.J. Collins. *Introduction to multivariate analysis*. Chapman & Hall, London, 1980.

H. Choi and S. Choi. Kernel isomap. *Electronics Letters*, 40, 2004.

R.R. Coifman, S. Lafon, A.B. Lee, M. Maggioni, B. Nadler, F. Warner, and S.W. Zucker. Geometric diffusions as a tool for harmonic analysis and structure definition of data: Multiscale methods. *Proc. Natl. Acad. Sci. USA*, 102(21):7433–7437, 2005.

J.A. Costa and A. Hero. Geodesic entropic graphs for dimension and entropy estimation in manifold learning. *IEEE Transactions on Signal Processing*, 25:2210–2221, 2004.

T. F. Cox and M. A. A. Cox. *Multidimensional Scaling*. Chapman & Hall, 1994.

N. Cristianini and J. Shawe-Taylor. *An introduction to support vector machines : and other kernel-based learning methods*. Cambridge University Press, 2000.

K. Dawson, R. Rodriguez, and W. Malyj. Sample phenotype clusters in high-density oligonucleotide microarray data sets are revealed using Isomap, a nonlinear algorithm. *BMC Bioinformatics*, 6(1):195, 2005.

H. de Jong. Modeling and simulation of genetic regulatory systems: A litterature review. *J. of computational biology*, 9(1):67–103, 2002.

V. de Silva and J.B. Tenenbaum. Global versus local methods in nonlinear dimensionality reduction. In *Advances in Neural Information Processing Systems 15*, 2003.

K.I. Diamantaras and S.Y. Kung. *Principal Component Neural Networks*. Wiley, New York, 1996.

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

D.L. Donoho and C. Grimes. Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. *PNAS*, 100(10):5591–5596, 2003.

P.G. Doyle and J.L. Snell. *Random walks and electric networks*. The Mathematical Association of America, 2000.

S. Dudoit and J. Fridlyand. Classification in microarray experiments. In T. Speed, editor, *Statistical analysis of gene expression microarray data*. Chapman & Hall/CRC, 2003.

M.B. Eisen, P.T. Spellman, P.O. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. *Proc. Natl. Acad. Sci. USA*, 1998.

A.L. Epstein and H.S. Kaplan. Feeder layer and nutritional requirements for the establishment and cloning of human malignant lymphoma cell lines. *Cancer Res.*, 39:1748–1759, 1979.

D. Fambrough, K. McClure, A. Kazlauskas, and E.S. Lander. Diverse signaling pathways activated by growth factor receptors induce broadly overlapping, rather than independent, sets of genes. *Cell*, 97:727–741, 1999.

E.W. Forgy. Cluster analysis of multivariate data: efficiency vs interpretability of classifications. *Biometrics*, 21:768–769, 1965.

T.S. Furey, N. Cristianini, N. Duffy, D.W. Bednarski, M. Schummer, and D. Haussler. Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics*, 2000.

M.E. Garber, O.G. Troyanskaya, K. Schluens, S. Petersen, Z. Thaesler, M. Pacyna-Gengelbach, M. van de Rijn, G.D. Rosen, C.M. Perou, R.I. Whyte, R.B. Altman, P.O. Brown, D. Botstein, and I. Petersen. Diversity of gene expression in adenocarcinoma of the lung. *Proc. Natl. Acad. Sci. USA*, 98(24):13784–13789, 2001.

T.R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, and E. S. Lander. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 1999.

M. Gustafsson, M. Hörnquist, and A. Lombardi. Constructing and analyzing a large-scale gene-to-gene regulatory network — lasso-constrained inference and biological validation. *IEEE/ACM Transactions on computational biology and bioinformatics*, 2, 2005.

J. Ham, D.D. Lee, S. Mika, and B. Schölkopf. A kernel view of the dimensionality reduction of manifolds. In *Proceedings of the 21st international conference on Machine Learning*, 2004.

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

X. He, S. Yan, Y. Hu, P. Niyogi, and H.-J. Zhang. Face recognition using laplacianfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27:328–340, 2005.

I. Hedenfalk, D. Duggan, Y. Chen, M. Radmacher, M. Bittner, R. Simon, P. Meltzer, B. Gusterson, M. Esteller, O.-P. Kallioniemi, B. Wilfond, Å. Borg, and J. Trent. Gene-expression profiles in hereditary breast cancer. *The New England Journal of Medicine*, February 2001.

M. Hein, J.-Y. Audilbert, and U. von Luxburg. From graphs to manifolds - weak and strong pointwise consistency of graph laplacians. In P. Auer and R. Meir, editors, *Learning theory*, volume 3559 of *Lecture notes in computer science*. Springer-Verlag, 2005.

G. E. Hinton and S. Roweis. Stochastic neighbor embedding. In *Advances in Neural Information Processing Systems, 15*, 2003.

N.S. Holter, A. Maritan, M. Cieplak, N.V. Fedoroff, and J.R. Banavar. Dynamic modeling of gene expression data. *Proc. Natl. Acad. Sci. USA*, 2001.

H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24:417–441, 498–520, 1933.

H. Hotelling. Relations between two sets of variates. *Biometrika*, 28:321–377, 1936.

J.Z. Huang, W.G. Sanger, T.C. Greiner, L.M. Staudt, D.D. Weisenburger, D.L. Pickering, J.C. Lynch, J.O. Armitage, R.A. Warnke, A.A. Alizadeh, I.S. Lossos, R. Levy, and W.C. Chan. The t(14;18) defines a unique subset of diffuse large B-cell lymphoma with a germinal center B-cell gene expression profile. *Blood*, 99:2285–2290, 2002.

R.A. Irizarry, B. Hobbs, F. Collins, Y.D. Beazer-Barclay, K.J. Antonellis, U. Scherf, and T.P. Speed. Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostatistics*, 4:249–264, 2003.

V.R. Iyer, M.B. Eisen, D.T. Ross, G. Schuler, T. Moore, J.C.F. Lee, J.M. Trent, L.M. Staudt, J. Hudson Jr., M.S. Boguski, D. Lashkari, D. Shalon, D. Botstein, and P.O. Brown. The transcriptional program in the response of human fibroblasts to serum. *Science*, 283:83–87, 1999.

V. Jain and L.K. Saul. Exploratory analysis and visualization of speech and music by locally linear embedding. In *Proceedings of the International Conference of Speech, Acoustics, and Signal Processing*, 2004.

T.K. Jenssen, A. Laegreid, J. Komorowski, and E. Hovig. A literature network of human genes for high-throughput analysis of gene expression. *Nature Genetics*, 28:21–28, 2001.

I. T. Jolliffe. *Principal Component Analysis*. Springer, 1986.

M. Kanehisa, S. Goto, M. Hattori, K.F. Aoki-Kinoshita, M. Itoh, S. Kawashima, T. Katayama, M. Araki, and M. Hirakawa. From genomics to chemical genomics: new developments in KEGG. *Nucl. Acids Res.*, 34:D354–357, 2006.

S.A. Kauffman. Homeostasis and differentiation in random genetic control networks. *Nature*, 224:177–178, 1969.

S.A. Kauffman. *The Origins of order: Self-organization and selection in evolution*. Oxford University Press, 1993.

L. Kaufman and P.J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley, New York, 1990.

J. Khan, R. Simon, M. Bittner, Y. Chen, S.B. Leighton, T. Pohida, P.D. Smith, Y. Jiang, G.C. Gooden, J.M. Trent, and P.S. Meltzer. Gene expression profiling of alveolar rhabdomyosarcoma with cDNA microarrays. *Cancer Research*, November 1998.

J. Khan, J.S. Wei, M. Ringnér, L.H. Saal, F. Westermann, F. Berthold, M. Schwab, C.R. Antonescu, C. Peterson, and P.S. Meltzer. Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nature Medicine*, 2001.

J. Khan, L.H. Saal, M.L. Bittner, Y. Jiang, G.C. Gooden, A.A. Glatfelter, and P.S. Meltzer. Gene expression profiling in cancer using cDNA microarrays. *Methods Mol. Med.*, 68: 205–222, 2002.

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

R. Küppers, U. Klein, M.-L. Hansmann, and K. Rajewsky. Mechanisms of disease: Cellular origin of human B-cell lymphomas. *New Engl. J. Med.*, 341:1520–1529, 1999.

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

S. Lafon. *Diffusion maps and geometric harmonics*. PhD thesis, Yale University, 2004.

R. Larsen. Non-linear shape decomposition using ISOMAP. In *9th Scnadinavian Conference on Chemometrics*. University of Iceland, 2005.

N.A. Laskaris and A.A. Ioannides. Semantic geodesic maps: a unifying geometrical approach for studying the structure and dynamics of single trial evoked responses. *Clinical Neurophysiology*, 113:1209–1226, 2002.

J.A. Lee, A. Lendasse, and M. Verleyseny. Curvilinear distance analysis versus isomap. In *European Symposium on Artificial Neural Networks*, 2002.

W. Li and C.H. Wong. Model-based analysis of oligonucleotide arrays: Expression index computation and outlier detection. *Proc. Natl. Acad. Sci. USA*, 98:31–36, 2001a.

W. Li and C.H. Wong. Model-based analysis of oligonucleotide arrays: model validation, design issues and standard error application. *Proc. Natl. Acad. Sci. USA*, 2, 2001b.

I.S. Lim, P. de Heras Ciechomski, S. Sarni, and D. Thalmann. Planar arrangement of high-dimensional biomedical data sets by isomap coordinates. In *Proceedings of the 16th IEEE Symposium on Computer-Based Medical Systems (CBMS'03)*, 2003.

C.-Y. Liou and Y.-T. Kuo. Economic states on neuronic maps. In *Proceedings of the 9th International Conference on Neural Information Processing (ICONIP'OZ)*, 2002.

D.J. Lockhart, H. Dong, M.C. Byrne, M.T. Follettie, M.V. Gallo, M.S. Chee, M. Mittmann, C. Wang, M. Kobayashi, H. Horton, and E.L. Brown. Expression monitoring by hybridization to high-density oligonucleotide arrays. *Nat. Biotechnol.*, 14:1675–1680, 1996.

J. Luo, D.J. Duggan, Y. Chen, J. Sauvageot, C.M. Ewing, M.L. Bittner, J.M. Trent, and W.B. Isaacs. Human prostate cancer and benign prostatic hyperplasia: molecular dissection by gene expression profiling. *Cancer Res.*, 61:4683–4688, 2001.

S.W. Malone, P. Tarazaga, and M.W. Trosset. Better initial configurations for metric multidimensional scaling. *Computational Statistics and Data Analysis*, 41:143–156, 2002.

S. Mehra, H. Messner, M. Minden, and R.S.K. Chaganti. Molecular cytogenetic characterization of non-hodgkin lymphoma cell lines. *Genes Chromosomes Cancer*, 33:225–234, 2002.

D. Mekarnia, A. Bijaoui, C. Delle Luche, and J.P. Maillard. Analysis of spectro-imaging data using the karhunen-loeve expansion: Application to ngc 7027 in the infrared. *Astronomy & Astrophysics*, 418:771–780, 2004.

S. Mika, B. Schölkopf, A. J. Smola, K.-R. Müller, M. Scholz, and G. Rätsch. Kernel pca and de-noising in feature spaces. In *Advances in Neural Information Processing Systems 11*, 1999.

B. Nadler, S. Lafon, R.R. Coifman, and I.G. Kevrekidis. Diffusion maps, spectral clustering and the reaction coordinates of dynamical systems. Technical report, Yale University, November 2004.

S. Nakauchi. Spectral imaging technique for visualizing the invisible information. In *Proceedings of the 14th Scandinavian conference on image analysis (SCIA'05)*, 2005.

A.Y. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems 14*, 2002.

J. Nilsson and F. Andersson. Circuit models for manifold learning. *submitted*, 2006.

J. Nilsson, T. Fioretos, M. Höglund, and M. Fontes. Approximate geodesic distances reveal biologically relevant structures in microarray data. *Bioinformatics*, 20(6):874–880, April 2004.

Y. Pawitan, S. Michiels, S. Koscielny, A. Gusnanto, and A. Ploner. False discovery rate and sample size for microarray studies. *Bioinformatics*, 21:3017–3024, 2005.

K. Pearson. Principal components analysis. *The London, Edinburgh and Dublin Philosophical Magazine and Journal*, 6:566, 1901.

N. Pochet, F. De Smet, J.A.K. Suykens, and B.L.R. De Moor. Systematic benchmarking of microarray data classification: assessing the role of non-linearity and dimensionality reduction. *Bioinformatics*, 20:3185–3195, 2004.

J.R. Pollack, M. Van de Rijn, and D. Botstein. Challenges in developing a molecular characterization of cancer. *Semin. Oncol.*, 29:280–285, 2002.

J Quackenbush. Computational analysis of microarray data. *Nature*, 2:418–427, 2001.

S. Ramaswamy, P. Tamayo, R. Rifkin, S. Mukherjee, C.H. Yeang, M. Angelo, C. Ladd, M. Reich, E. Latulippe, J.P. Mesirov, T. Poggio, W. Gerald, M. Loda, E.S. Lander, and T.R. Golub. Multiclass cancer diagnosis using tumor gene expression signatures. *Proc. Natl. Acad. Sci. USA*, 98:15149–15154, 2001.

M.E. Ross, X. Zhou, G. Song, S.A. Shurtleff, K. Girtman, W.K. Williams, H-C Liu, R. Mahfouz, S.C. Raimondi, N. Lenny, A. Patel, and J.R. Downing. Classification of pediatric acute lymphoblastic leukemia by gene expression profiling. *Blood*, 102: 2951–2959, 2003.

F.P. Roth, J.D. Hughes, P.W. Estep, and G.M. Church. Finding dna regulatory motifs within unaligned non-coding sequences clustered by whole-genome mrna quantitation. *Nature Biotechnol.*, 16:939–945, 1998.

S.T. Roweis and L.K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 2000.

J.W. Sammon Jr. A non-linear mapping for data structure analysis. *IEEE Transactions on computers*, 18:401–409, 1969.

M. Schena, D. Shalon, R.W. Davis, and P.O. Brown. Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science*, 270:467–470, 1995a.

M. Schena, D. Shalon, R. Heller, A. Chai, P.O. Brown, and R.W. Davis. Parallel human genome analysis: microarray-based expression monitoring of 1000 genes. *Proc. Natl. Acad. Sci. USA*, 93:10614–10619, 1995b.

S.S. Schiffman, M.L. Reynolds, and F.W. Young. *Introduction to Multidimensional Scaling*, pages 362–371. Academic Press, New York, 1981.

B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998.

B. Schölkopf, S. Mika, C.J.C. Burges, P. Knirsch, K.-R. Müller, G. Rätsch, and A. J. Smola. Input space versus feature space in kernel-based methods. *IEEE Transactions on Neural Networks*, 10:1000–1017, 1999.

P. Sebastiani, E. Gussoni, I.S. Kohane, and M.F. Ramoni. Statistical challenges in functional genomics. *Statistical Science*, 18:33–70, 2003.

C. Shi and C. Lihui. Feature dimension reduction for microarray data analysis using locally linear embedding. In *Proceedings of 3rd Asia-Pacific Bioinformatics Conference*, 2005.

R. Sibson. Order invariant methods for data analysis. *Journal of the Royal Statistical Society, series B*, 34:311–349, 1972.

A.J. Smola, R.C. Williamson, S. Mika, and B. Schölkopf. Regularized principal manifolds. *Lecture Notes in Computer Science*, 1572:214–229, 1999.

J. D. Storey and R. Tibshirani. Statistical significance for genomewide studies. *Proc. Natl. Acad. Sci. USA*, 100:9440–9445, 2003.

M. Åstrand. Contrast normalization of oligonucleotide arrays. *Journal of Computational Biology*, 10:95–102, 2003.

A. Subramanian, P. Tamayo, V.K. Mootha, S. Mukherjee, B.L. Ebert, M.A. Gillette, A. Paulovich, S.L. Pomeroy, T.R. Golub, E.S. Lander, and J.P. Mesirov. Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *Proc. Natl. Acad. Sci. USA*, 102:15545–15550, 2005.

P. Tamayo, D. Slonim, J. Mesirov, Q. Zhu, S. Kitareewan, E. Dmitrovsky, E.S. Lander, and T.R. Golub. Interpreting patterns of gene expression with self-organizing maps: Methods and application to henatopoietic differentiation. *Proc. Natl. Acad. Sci. USA*, 1999.

S. Tavazoie, J.D. Hughes, M.J. Campbell, R.J. Cho, and G.M. Church. Systematic determination of genetic network architecture. *Nature Genetics*, 22:281 – 285, 1999.

J.B. Tenenbaum, V. de Silva, and J.C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2322, 2000.

R. Tibshirani. Regression shrinkage and selection via the lasso. *J. Royal Statistical Soc. Series B*, 58:267288, 1996.

W.S. Torgerson. Multidimensional scaling I. *Psychometrika*, 17:401–419, 1952.

O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R.B. Altman. Missing value estimation methods for dna microarrays. *Bioinformatics*, 17:520–525, 2001.

M.E. Tweeddale, B. Lim, N. Jamal, J. Robinson, J. Zalcberg, G. Lockwood, M.D. Minden, and H.A. Messner. The presence of clonogenic cells in high-grade malignant lymphoma: a prognostic factor. *Blood*, 69:1307–1314, 1987.

E.P. van Someren, L.F.A. Wessels, and M.J.T. Reinders. Linear modeling of genetic networks from experimental data. *Proc. Int. Conf. Intell. Syst. Mol. Biol.*, 2000.

J. Venna and S. Kaski. Visualized atlas of a gene expression databank. In *Proceedings of Symposium of Knowledge Representation in Bioinformatics*, 2005.

J.P. Vert and M. Kanehisa. Extracting active pathways from gene expression data. *Bioinformatics*, 19:238–244, 2003.

U. von Luxburg, M. Belkin, and O. Bousquet. Consistency of spectral clustering. Technical Report TR-134, Max-Planck-Institut für biologische Kybernetik, 2004.

K.Q. Weinberger, F. Sha, and L.K. Saul. Learning a kernel matrix for nonlinear dimensionality reduction. In *Proceedings of the Twenty First International Confernence on Machine Learning (ICML-04)*, 2004.

C.K.I. Williams. On a connection between kernel pca and metric multidimensional scaling. *Machine Learning*, 46:11–19, 2002.

H. Wold. Estimation of principal components and related models by iterative least squares. In *Multivariate Analysis*. Academic Press, NY, 1966.

Y.H. Yang, S. Dudoit, P. Luu, D.M. Lin, V. Peng, J. Ngai, and T.P. Speed. Normalization for cDNA microarray data: a robust composite method addressing single and multiple slide systematic variation. *Nucleic Acids Research*, 30, 2002.

M.K.S. Yeung, J. Tegner, and J.J. Collins. Reverse engineering gene networks using singular value decomposition and robust regression. *PNAS*, 99(9):6163–6168, 2002.

S. Zhong, C. Li, and W.H. Wong. ChipInfo: software for extracting gene annotation and gene ontology information for microarray analysis. *Nucl. Acids Res.*, 31(13):3483–3486, 2003.