INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

ProQuest Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

UMI®

DISSERTATION

MODELING GEOMETRIC STRUCTURE IN NOISY DATA

Submitted by

Markus Gerhard Anderle

Department of Mathematics

In partial fulfillment of the requirements

for the degree of Doctor of Philosophy

Colorado State University

Fort Collins, Colorado

Summer 2001

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

COLORADO STATE UNIVERSITY

May 18, 2001

WE HEREBY RECOMMEND THAT THE DISSERTATION PREPARED UNDER OUR SUPERVISION BY MARKUS GERHARD ANDERLE ENTITLED "MODELING GEOMETRIC STRUCTURE IN NOISY DATA" BE ACCEPTED AS FULFILLING IN PART REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY.

Committee on Graduate Work

Adviser

Department Head

ABSTRACT OF DISSERTATION

MODELING GEOMETRIC STRUCTURE IN NOISY DATA

We present an approach for modeling noisy data via dimension reduction methods. Geometric structures, hidden in the ambient space defined by the dimension of the observations, are uncovered by the application of efficient clustering algorithms, based on the exploitation of nearest neighbor interactions. A new bi-directional Hebb rule in combination with the LBG algorithm was used to define a connectivity structure among disjoint regions in high-dimensional space. For a lossless representation of noisy data the Whitney Reduction Network was combined with the maximum noise fraction filter to create a more accurate model of the underlying data generator while utilizing the set of unit secants in a sequential algorithm to construct a good quality parameterization of the data. The nonlinear reconstruction of the data was addressed by the feedback of a model validation test on the residuals to form a radial basis function resource allocation architecture.

Markus Gerhard Anderle
Department of Mathematics
Colorado State University
Fort Collins, Colorado 80523
Summer 2001

ACKNOWLEDGEMENTS

First of all, my sincere thanks and gratitude go to my supervisor, Dr. Michael Kirby, for his continuous guidance, inspiration and enthusiasm. Without his wealth of knowledge and extraordinary insight, expressed in many discussions, this dissertation would have not been possible.

I would also like to thank Dave Broomhead, Gerhard Dangelmayer, Rick Miranda, Doug Hundley, Sabino Gadaleta, and Anthony Toad for their many helpful discussions and explanations during the preparation of this dissertation.

I also want to thank the Department of Mathematics at Colorado State University for their hospitality.

Finally, I am forever indebted to my parents for their continuous support and generosity.

This research was supported in part by the NSF under grant DMS-9973303 and Honeywell, Inc.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

The modeling and understanding of "real-world" data poses a contemporary challenge to many intelligent methodologies developed in the framework of neural networks, pattern recognition, statistics, etc. The common goal is often to maximize the extraction of knowledge about the generator of the data, including engineering processes, imagery, climate changes, financial markets, even humans, providing us with a whole array of complex biomedical signals, e.g., Electrocardiogram (ECG) or Electroencephalogram (EEG) recordings. The world wide web has contributed to the ease of access and distribution of enormous amounts of data, and may also be regarded as a generator of an immense volume of information, both in text and numeric form. In many cases it may be even critical to have an accurate model of the observed signal, particularly if some kind of abnormal mode of operation is encountered, e.g., the failure of a component in a chemical process, which makes an immediate intervention necessary.

The main methodology used in this dissertation to model and to discover structure in data is based on a geometric approach to *dimension reduction mappings*.

1.1 Geometric Dimension Reduction

Assume we are provided with a signal in the form of a large number P of numeric q-tuple samples of the process under investigation. With the construction

of an appropriate compression mapping we are able to perform a *lossy* dimension reduction, effectively reducing the amount of samples P of a given data set. Here, we lose the ability to reconstruct the original data, but gain an approximation which is less complex. This may be accomplished, for example by a *clustering* or *vector quantization technique*, that associates prototypes with actual data points.

In a *lossless* compression of the data, we seek a reduction *and* a perfect reconstruction mapping, necessarily leaving the actual dimension of the reduced data representation unchanged. Utilizing the geometric approach to data modeling we assume that the data resides on a geometric object embedded in a larger vector space, that we are able to describe mathematically. For the discovery of the geometric structure of the data we will employ *dimension reduction networks*.

1.1.1 Lossy Dimension Reduction Using Clustering

Here, the data is "squashed" by choosing M prototypes, or cluster centers, as representatives, effectively replacing the entire data set, where $M \ll P$. Such a reduced representation is necessary as a preprocessing step in many modeling techniques, e.g., as an initial center set for Gaussian mixture modeling as a tool to approximate the underlying probability distribution of the data [9], as well as in the context of nonlinear function approximation via radial basis function neural networks [11]. Current problems encountered in constructing a cluster representation using large amounts of data are the following.

- The computational bottlenecks associated with the processing of enormous amounts of data require the development of fast and efficient algorithms.
- The exploitation of the sparsity of data located in disjoint regions in high-dimensional space. In general, the geometric structure of the data set should be incorporated into current clustering algorithms.

- A fundamental problem in many fields of study is the need to organize large data sets and detect useful groupings or clusters, and to establish relations among them; a task usually encountered in the context of *data mining* [15].

The first chapter of this thesis will show how a successful clustering provides us with a mapping that uncovers structural relationships hidden in the data distribution, quantized by some kind of similarity measure, while pruning as many computational operations as possible. The novel approaches introduced in this dissertation addressing the challenges listed above include[1]:

- ◇ the introduction of a bi-directional Hebb rule to discover hidden states of the data, while using only a small set of cluster centers;
- ◇ quantifying local neighborhoods and their exploitation for accelerating clustering algorithms;
- ◇ a rule-based relocation algorithm of cluster centers based on local neighborhood interactions to increase the efficiency of a given set of cluster centers.

1.1.2 Lossless Dimension Reduction Using Networks

The dimension of the input sensory data might be very high, whereas the actual physical phenomenon may be governed only by a few parameters. The task is now to uncover these *latent variables* or *hidden states* and represent the actual observations using a lower dimensional model. It is often possible to reduce the dimension of the data due to the fact that the observables of the system under investigation contain irrelevant measurements and represent correlations between the hidden states of the process. In addition many measurements contain high

[1]Current problems associated with a method are presented in a •-list. The proposed solutions this dissertation presents are summarized in a ◇-list.

levels of noise. In a geometric approach we would separate the noise from the interesting signal via their spatial geometry, i.e., their separation into noise and signal subspaces. An accurate model of the process might then exclude hidden states associated with noise.

Geometrically, hidden states form an object such as a vector subspace or a submanifold, and it is the dimension d of this object that serves us in reducing the dimension of the data. The original dimension, referred to as the *ambient dimension* q is naturally given by the observed q-tuple, i.e., the number of components of a multivariate measurement. In a practical, or empirical approach to discover the geometric structure of the hidden states, the *intrinsic dimension* d, is taken to be the minimum number of hidden states or parameters needed to describe the model accurately. We try here to construct empirical reduction mappings such that $d \ll q$. Denoting the reduction mapping by $G : U \subset \mathbb{R}^q \rightarrow V \subset \mathbb{R}^d$ and the reconstruction mapping as $H : V \subset \mathbb{R}^d \rightarrow U \subset \mathbb{R}^q$ we may write symbolically [39],

$$U \xrightarrow{G} V \xrightarrow{H} U, \tag{1.1}$$

where the composite mapping produces the identity for $u \in U$

$$u = (H \circ G)(u). \tag{1.2}$$

1.1.3 Current Dimension Reduction Methods

The computational implementation of Equation (1.1) for the empirical estimation of G and H via a data set may be accomplished by an *autoassociator* [41]. If linear functions H, G are employed for the reduction and reconstruction mapping, the autoassociator implements a compression scheme known as PCA [5]. If nonlinear functions G, H are used the networks becomes a *bottleneck network* or a nonlinear autoassociator. In the framework of neural networks the nonlinear

functions for the reduction mapping G and its inverse H are implemented via multilayer neural networks. Despite several drawbacks associated with the nonlinear autoassociator architecture, such as the non-uniqueness of the mappings G and H, and slow network training if large hidden layers are used, the nonlinear autoassociators are a popular tool, e.g., to discover hidden states in a chemical process for the propose of fault diagnosis [37].

Another tool for the discovery of the intrinsic structure of observed data is Projection Pursuit (PP) [22][34]. It can be distinguished from our geometric approach in that it utilizes purely the underlying probabilistic structure of the data. The goal of PP is to discover *interesting* low-dimensional orthogonal projections by optimizing a objective function called *projection index*. Interesting projections are, for example, found via information-theoretic measures such as negative entropy [19]; in this sense an interesting projection discovers directions which produces distributions that are least normal. In its extension, PP Regression [22] fits the original data employing a nonlinear reconstruction.

Some disadvantages associated with these current methods include:

- the intrinsic dimension d is not determined by the architecture directly, and often a result of trail-and-error runs;
- the conditioning of the mappings G and H are not optimized by the training procedures, and the resulting network may exhibit bad generalization properties.

The Whitney Reduction Network approaches these problems using a geometric approach to dimensionality reduction.

1.1.4 The Whitney Reduction Network

The method employed in this thesis for the nonlinear dimension reduction and the discovery for the hidden states of the process is the Whitney Reduction

Network (WRN) [13, 12, 39]. In the WRN, the dimension reduction function G consists of a linear orthogonal projection, used to provide a low dimensional parameterization of the data. The inverse mapping H for the reconstruction is composed of the linear inverse of the projector plus a nonlinear reconstruction.

Based on *Whitney's Embedding Theorem* [33] the following reduction and reconstruction architecture (see Figure 1.1) provides a decomposition of $x \in U \subset \mathbb{R}^q$, residing in the ambient space, and can be summarized in the following three steps.

Step 1: Decomposition of x into the range and null space of a projector $\mathbb{P}$

$$x = \mathbb{P}x + \mathbb{Q}x, \text{ where } \mathbb{Q} = I - \mathbb{P} \tag{1.3}$$

$$= p + q \tag{1.4}$$

$$= p + f(p). \tag{1.5}$$

Whitney's Embedding Theorem ensures the existence of a global map $q = f(p)$, given that $d > 2m$, where d is the rank of $\mathbb{P}$ and m the dimension of the manifold on which the data resides.

Step 2: Parameterization of x with projector $\mathbb{P}$ in basis $V = [v_1|\ldots|v_q]$. The projectors are given via splitting the basis into $V_1 = [v_1|\ldots|v_d]$ and $V_2 = [v_{d+1}|\ldots|v_q]$

$$\mathbb{P} = V_1 V_1^T \text{ and } \mathbb{Q} = V_2 V_2^T,$$

the d-dimensional parameterization of x is given as $\hat{p} = V_1^T x$ and $\hat{q} = V_2^T x$.

The main feature of the WRN is the construction of a projector $\mathbb{P}$, providing a parameterization of the data, such that the inverse mapping is as well conditioned as possible, using the notion of *good projections* [13, 12]. A well conditioned mapping will not amplify small perturbations in the domain of the nonlinear function approximation; this property is essential, if we seek a dimension reduction that

generalizes well to data not contained in the original set. As a result, this dimension reduction network will also exhibit better approximation properties in the case where the observed signal is contaminated with a noise process. The WRN architecture will therefore extract the number of hidden states (dimension d) such that it is possible to construct an accurate model. Despite the theoretical foundation based on the Whitney's Embedding Theorem, the practical implementation of the first and second WRN step are associated with the following problems.

- The quest for a good projection is dictated by the trade-off between a low dimensional parameterization and a well-conditioned inverse, possibly leading to conflicting objectives during the optimization of the WRN.
- Finding a good projection is considerably influenced by the amount of noise present in the observations. The current noise-adjusted WRN employs an *ad hoc*, user-supplied cutoff to extract a good projection.

We will utilize the *set of secants* to construct good quality projections, which is the topic of Chapter 4, introducing

- ◇ the maximum noise fraction method for noise filtering of multivariate signals;
- ◇ a local secant algorithm, based on a partitioning of the data;
- ◇ The sequential secant adaption algorithm for computing good projections.

The third step in the WRN employs a nonlinear function approximation method to fit the nonlinear portion of the reconstruction.

> Step 3: Reconstruction of x from $\hat{p}$ and an approximation of $\hat{q} = f(\hat{p})$, utilizing a nonlinear function fit to $f(\hat{p})$ results in $\tilde{f}(\hat{p}) : \hat{p} \rightarrow \tilde{\hat{q}} = \tilde{f}(\hat{p})$ followed by the reconstruction to the ambient space
>
> $$p = \hat{U}_1 \hat{p} \text{ and } \tilde{q} = \hat{U}_2 \tilde{\hat{q}},$$

where (1.2) is approximated as $I \approx \tilde{\mathbb{P}}^{-1} \circ \mathbb{P} : x \rightarrow \tilde{x}$:

$$\tilde{x} = \hat{U}_1 \hat{p} + \hat{U}_2 \tilde{f}(\hat{p}).$$

The problems encountered in Step 3 are associated with optimizing the parameters of a nonlinear model from empirical, noisy data and include:

- the choice of the appropriate model order is essential in avoiding overfitting and bad generalization;
- many current, neural-network-based, function approximations require user-supplied *ad hoc* parameters that dictate the quality of the resulting fit;
- many radial basis function neural networks, when employed for fitting a nonlinear relationship, require a data representation in the form of cluster centers.

In Chapter 2 we propose to utilize a *model validation test* [8], a common tool in systems identification [46], to control the size of an RBF network. Under the assumption of additive iid noise, we are able to avoid overfitting by reproducing the correct statistics on the residuals without the requirement of a cross-validation set to terminate the training procedure. The implementation of this idea leads to the

- ◇ Autocorrelation Feedback Radial Basis Function Network.

In addition, the clustering algorithms presented in Chapter 2 may be employed to the center selection problem encountered in radial basis function networks.

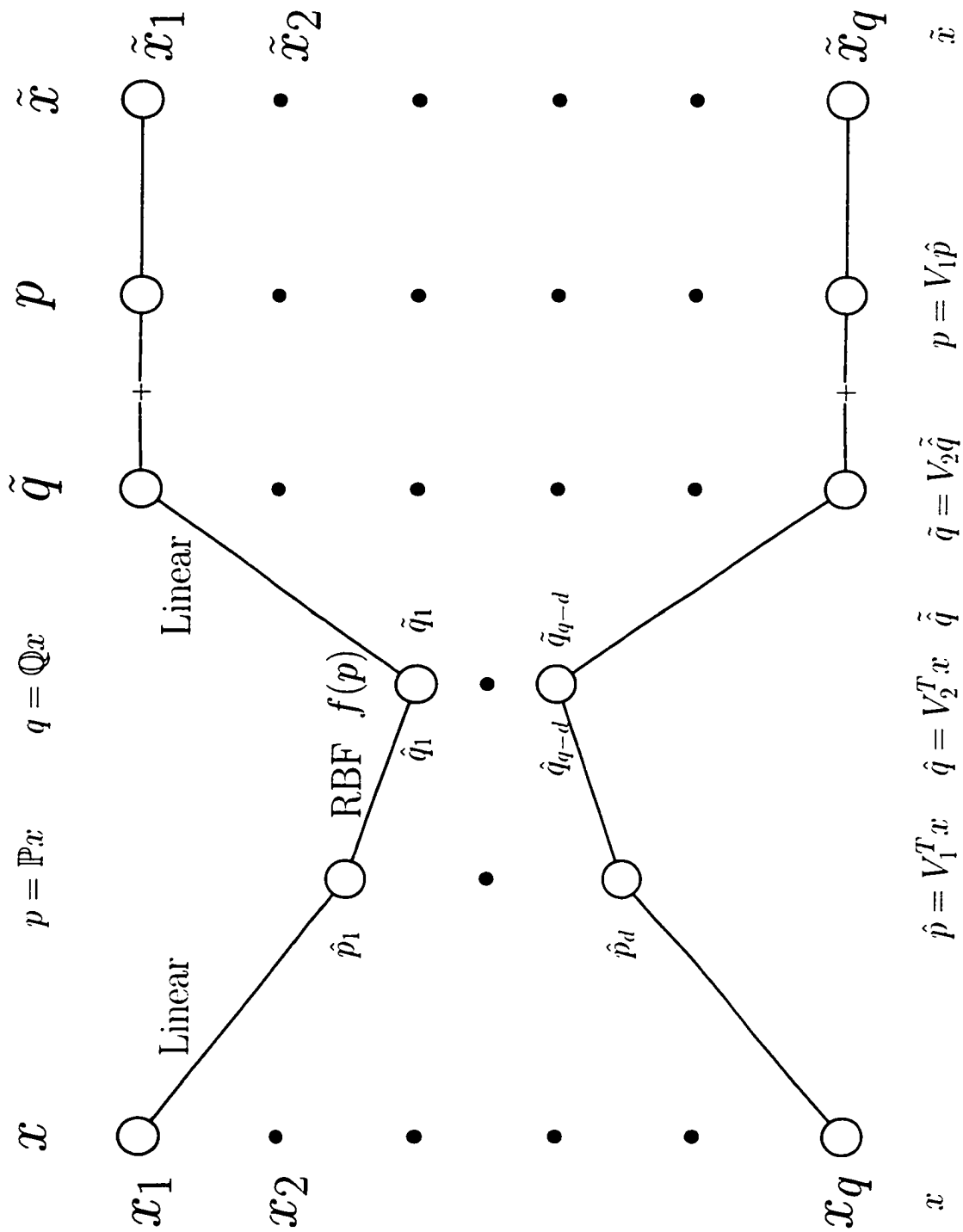


Figure 1.1: The Whitney Reduction Network architecture. Nodes denoted by • are placeholders for additional nodes, depending on the dimension of the data.

Chapter 2

CLUSTERING

In this chapter we present and discuss LBG (Linde-Buzo-Gray) [45] and k-means [21, 50] based clustering algorithms and investigate interactions between clusters and data points. A bi-directional Hebbian learning rule is proposed that constructs a set of uncoupled Delauney Triangulations of the data and identifies regions that are occupied by high-dimensional data. Local interactions between data and center points are used to reduce the necessary computational power for clustering algorithms and increases their overall efficiency.

2.1 Introduction

Clustering algorithms aim to process data in such a way that inherent and hidden similarity structure is revealed. Grouping, or organizing, the data into regions, or clusters, is central in knowledge extraction, data modeling and pre-processing (e.g., Expectation-Maximization [9]). Processing large data sets by associating a prototype value with its nearest neighbors is a simple and appealing approach to data compression and classification and a standard method in data mining. This idea, referred to generically as clustering or vector quantization, may be implemented by a variety of algorithms including k-means [21, 50] and LBG [45] . In the last decade a number of neural based algorithms have been proposed which extend the basic goal of vector quantization to data reduction with either self-organizing or topology preserving features including Kohonen's mapping and

neural gas [40, 51]. In order to avoid the problematic selection of *ad hoc* learning parameters we chose to work with batch algorithms, of which k-means and LBG are the most dominant members. Despite the basic local nature of the information which dictates the arrangement or selection of the cluster centers, these algorithms are essentially global in nature, implementing exhaustive searches over all data points and prototpyes [2].

To construct a vector quantization of a data set $\mathcal{X}$ consisting of P vectors $\{x^{(\mu)}, \mu = 1..P\}$, each in $\mathbb{R}^n$, we seek a mapping

$$Q : \mathbb{R}^n \rightarrow \mathcal{C},$$

$$Q(x) = c_i, \text{ if } x \in V_i,$$

and an associated membership function of x to some cell or class V_i:

$$m_i(x) = \begin{cases} 1, & x \in V_i \\ 0, & x \notin V_i \end{cases} \quad x \in \mathcal{X}. \tag{2.1}$$

In a sequential approach we denote by $m_i^{(t)}(x)$ the membership function of point x at iteration t. The case of a binary or hard membership function (Equation (2.1)) is also known as hard clustering, while soft clustering permits the membership of a data point to multiple clusters, e.g., fuzzy clustering. The compression results from the fact that the cardinality of the generated prototype or *cluster center* set $\mathcal{C} = \{c_1, \ldots, c_M\}$, $c_i \in \mathbb{R}^n$, $i = 1, .., M$, denoted by $M = |\mathcal{C}|$, is much less than the total number of patterns, i.e., $M \ll P$. With the use of the membership function $m(x)$, we may express $Q(x)$ as

$$Q(x) = \sum_{k=1}^{M} c_k m_k(x).$$

Let the index set $\mathcal{I}_1$ assign each $x \in \mathcal{X}$ the index of an associated cluster center

$$\mathcal{I}_1(x) = \sum_{k=1}^{M} m_k(x) k.$$

The indices for second nearest centers are contained in the index set $\mathcal{I}_2$ for further reference.

The functional that measures the quality of using the cluster center set $\mathcal{C}$ instead of the raw data is the distortion error. A common measure for distortion is the Euclidian distance, which for the entire quantization error is given by

$$E(\mathcal{X}, Q) = \sum_{x \in \mathcal{X}} \min_{c_i \in \mathcal{C}} ||x - c_i||^2 \tag{2.2}$$

$$= \sum_{x \in \mathcal{X}} ||x - Q(x)||^2. \tag{2.3}$$

Two conditions for a local minimum of $E(\mathcal{X}, Q)$ are known, but are only necessary for a global minimum [45]:

C1 Voronoi partition:

$$Q(x) = c_i \text{ if } x \in V_i \text{ , where}$$

$$V_i = \{x \in \mathbb{R}^n \mid ||x - c_i|| \leq ||x - c_k||,\ i \neq k, k = 1..M\}.$$

Given a set of cluster centers the cell partition must be a Voronoi partition (see Figure 2.1).

C2 Generalized Centroid assignment:

Given a cell partition the cluster centers must be the centroids of V_i

$$c_i = \frac{1}{|V_i|} \sum_{x \in V_i} x.$$

The computational implementations of C1 and C2 results in the LBG algorithm 2.1, which has been shown to converge to a local minimum of E [45].

In order to reveal the geometrical structure of the data represented by cluster centers and to solve proximity problems efficiently, the graph of the Delauney Triangulation [66] can be computed. It contains a (straight-line) edge connecting

Algorithm 2.1 LBG($\mathcal{C}, \mathcal{X}$)

Input: Initial (e.g. random) cluster centers $\mathcal{C}$, data $\mathcal{X}$
Output: $\mathcal{C}, \mathcal{I}_1$

```
1: Initialize old𝒞 ≠ 𝒞 {oldℐ1 ≠ ℐ1}
2: while 𝒞 ≠ old𝒞 {alternative: ℐ1 ≠ oldℐ1} do
3:    if old$c_i$ ≠ $c_i$ then
4:       Update $D(c_i, \mathcal{X})$ (see Equation (2.4))
5:    end if
6:    Sort for ℐ1 {Closest $c_i$ determine Voronoi cell $V_i$}
7:    $c_i = \frac{1}{|V_i|} \sum_{x \in V_i} x$
8:    old𝒞 ⇐ 𝒞 {oldℐ1 ⇐ ℐ1}
9: end while
```

two cluster centers c_i and c_j in the plane if and only if their Voronoi cells share a common edge. The connections are stored in an adjacency matrix $\mathcal{A}$:

$$\mathcal{A}_{ij} = \begin{cases} 1 & \text{if } V_i \cap V_j \neq \emptyset \quad V_i, V_j \text{ are adjacent} \\ 0 & \text{if } V_i \cap V_j = \emptyset \quad V_i, V_j \text{ are not adjacent} \end{cases}$$

Of special interest is a subgraph of the Delauney Triangulation, called "induced Delauney Triangulation", that produces Topology Preserving Maps (TPM)[40, 52]. It has been shown that the Competitive Hebbian Rule (CHR), under the rather strict condition of dense center distribution, results in such an "induced Delauney Triangulation":

> Competitive Hebbian Rule (CHR):
>
> For each input signal $x \in \mathcal{X}$ connect the two closest centers $\mathcal{I}_1(x)$ and $\mathcal{I}_2(x)$ (measured by some distance metric) by an edge [25].

This rule may also be implemented as a learning procedure [52]. According to Martinez and Schulten [52], a set of centers $\mathcal{C}$ is dense if it resolves the structural details of the data set, which can be achieved by choosing the number of centers $M = |\mathcal{C}|$ large. However, it is not clear how large M should be chosen, or if a clustering algorithm produces dense cluster centers. Geometrically, with the definition of 2nd order Voronoi cells,

$$V_{ij} = \{x \in \mathbb{R}^n \mid ||x - c_i|| \leq ||x - c_k|| \wedge ||x - c_j|| \leq ||x - c_k|| \; \forall k \neq i, j\},$$

the two closest units are connected according to the CHR if $\exists x \in \mathcal{X}$ s.t. $x \in V_{ij}$.

The application of the CHR to a test data set consisting of 1000 data points sampled from a Gaussian mixture distribution generated from 10 centers with diagonal covariance matrix is demonstrated[1] in Figure 2.2. We will show that the conditions under which two centers are connected should be more stringent in order to produce a stable connectivity structure, introducing a bi-directional Hebbian rule. Further we will show that the global computations necessary to update distances between $\mathcal{X}$ and $\mathcal{C}$ as well as the necessary sorts to determine $\mathcal{I}_1$ and $\mathcal{I}_2$ can be dramatically reduced if one considers that cluster center adaptation beyond some stage only results in local changes of memberships of data points that are elements of the set V_{ij} (see Figure 2.1).

The main task here is to define locality and appropriate neighborhoods that are involved in membership changes via 2nd order Voronoi cells and how to use them for efficiency improvement. This carries over to the fact that at late stages in the adaption process only local interactions between $\mathcal{X}$ and $\mathcal{C}$ are significant and most closeness evaluations are unnecessary.

2.2 Clustering Algorithms

In the following we present and discuss the LBG algorithm [45, 50] and a recent improvement called LBG-U [26]. The novel approach in the LBG-U design is the use of a utility-measure-based relocation of cluster centers that perform inadequatly. It therefore tries to overcome the heavy dependence of LBG on the initial cluster center distribution. We also include comparisons to one of the most re-

[1] All experiments were conducted on a Pentium III with 128 MB of RAM under Windows 98 in Matlab.

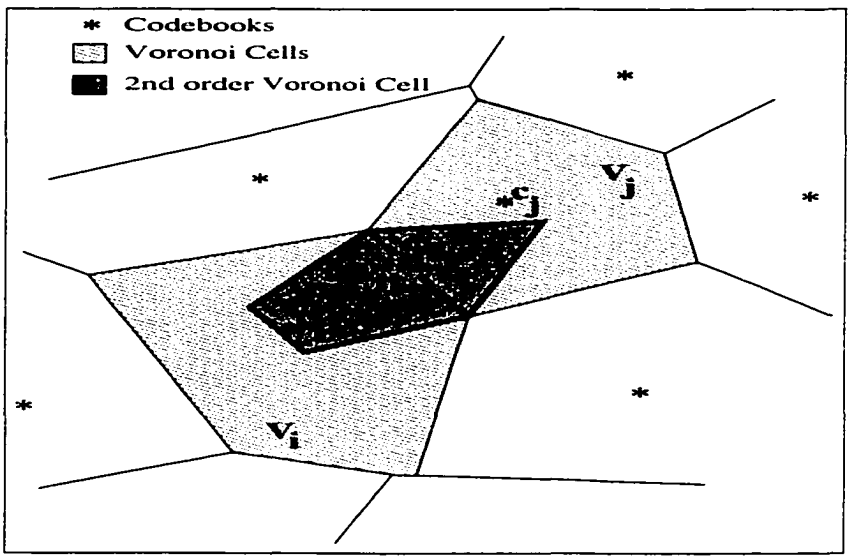


Figure 2.1: Voronoi cells V_i and V_j and second order Voronoi cell V_{ij}. After some iteration only points of V_j that are also in V_{ij} will change their membership to i. A complete distance update $D(c_i, \mathcal{X})$ to all points in $\mathcal{X}$ (as performed in LBG) is unnecessary.

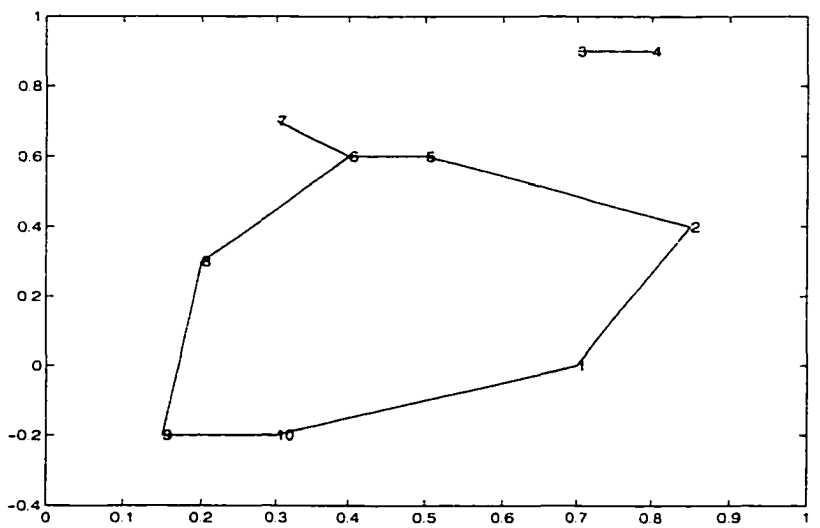


Figure 2.2: Connections established according to competitive Hebbian rule between 10 centers (numbers 1-10) from a Gaussian mixture.

cently introduced algorithms, generalized k-harmonic means (GKHM) [74], which claims to improve some of the limitations associated with the LBG algorithm.

2.2.1 The LBG Algorithm

The LBG Algorithm 2.1 implements the sufficient conditions C1 and C2 which require computationally expensive operations of order $O(M \times P \times n)$ per iteration to update an $M \times P$ distance matrix $D(\mathcal{C}, \mathcal{X})$ with

$$D(c_i, x_j) = D_{ij} = ||c_i - x_j||^2 \tag{2.4}$$

and the sorting for the closest center for each data point, stored in $\mathcal{I}_1$. In addition, one observes a strong dependence of the distortion error on the initial distribution of the cluster centers. Furthermore, after a fast initial convergence, the algorithm performs only small subsequent changes in the cluster center positions, which increases the number of iterations needed to terminate with essentially no center movement and improvement in distortion error (see plateau in Figure 2.3). This observation is due to the fact that at later iterations only a very small amount of data points change their center membership and therefore influence the center position only marginally. Several stopping criteria for the termination of the LBG algorithm are possible (see alternative termination criteria in Algorithm 2.1); setting a threshold for the maximum distance a center can move in consecutive iterations is also a possible termination criteria. In addition, we observe that data points change membership to a neighboring (connected) center after some iterations. Figure 2.4 (b) shows that no data points become members of centers other than second closest, after the second iteration.

We remark that the popularity of the LBG algorithm has prompted the development of faster and more efficient implementations. Among the most promising improvements appear to be k dimensional trees ($k-d$ trees) [59] and metric trees

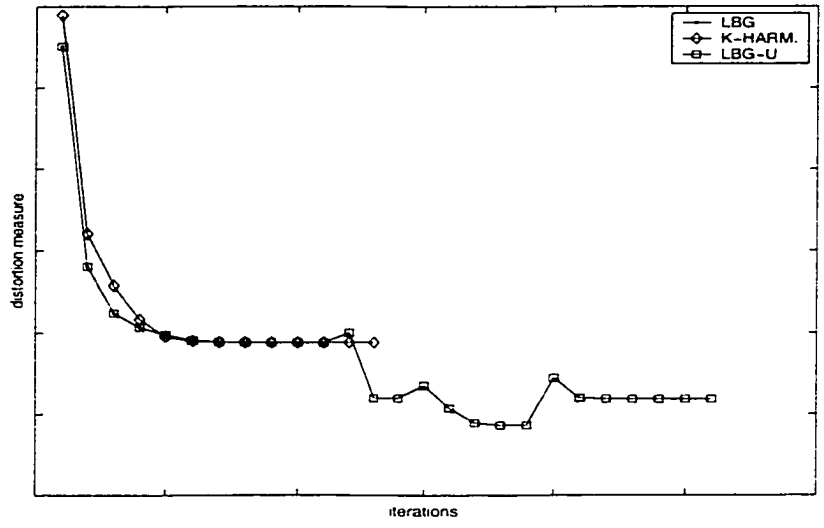


Figure 2.3: Typical performances of the LBG, LBG-U, and harmonic k-means algorithms.

[60] in the context of the Anchors Hierarchy. The methods developed here concern exploiting a notion of locality and should be useful in the context of methods based on intelligent computational structures, such as binary search trees [18].

2.2.2 Generalized K-Harmonic Means

Recently, a new iterative clustering procedure related to k-means based algorithms was introduced that addresses the sensitivity to the initialization of the centers. In the generalized k-harmonic means algorithm (GKHM) [74] the minimum distance in Equation (2.2) is replaced with the harmonic averages of the distances from the data points $x \in \mathcal{X}$ to all centers $c \in \mathcal{C}$. This replaces the winner-takes-all strategy of the LBG algorithm and makes GKHM a member of the fuzzy clustering family. In employing a harmonic average, the algorithm boosts data that are not close to any centers and gives them higher weight. The min function in Equation (2.2) is replaced with

$$\underset{x \in \mathcal{X}}{\mathrm{HA}}(||x - c||^2) = \frac{|\mathcal{C}|}{\sum_{c \in \mathcal{C}} \frac{1}{||x-c||^2}}. \tag{2.5}$$

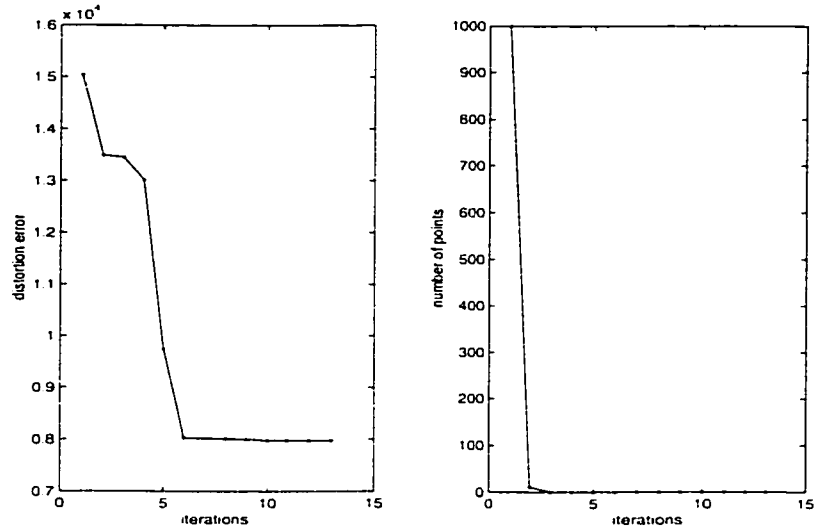


Figure 2.4: (a) Distortion error of the LBG algorithm (b) Number of points that do not change membership from $\mathcal{I}_1$ to $\mathcal{I}_2$ or $\mathcal{I}_2$ to $\mathcal{I}_1$.

A recursive formula for the center updates follows from the derivative of the k-harmonic distortion error

$$E_{\mathrm{GKHM}} = \sum_{\substack{x \in \mathcal{X} \\ c \in \mathcal{C}}} \mathrm{HA}(||x - c||^2), \tag{2.6}$$

with respect to the centers c_k:

$$c_k = \sum_{i=1}^{P} \frac{1}{D_{k,i}^4 (\sum_{j=1}^{M} \frac{1}{D_{j,i}^2})^2} x_i \Big/ \sum_{i=1}^{P} \frac{1}{D_{k,i}^4 (\sum_{j=1}^{M} \frac{1}{D_{j,i}^2})^2} \tag{2.7}$$

Line 7 in algorithm 2.1 is replaced with the iterative center update formula (2.7). Applied to the test data set, we observed only little performance improvement in terms of the distortion error, although GKHM is more stable with respect to a bad choice of the initial center distribution. Compared with the following LBG-U algorithm, the GKHM was not able to improve the positions of the cluster center vectors significantly. With about twice the number of iterations, GKHM was able to reduce the distortion error by 6% compared to LBG, averaged over 10000 runs (see Figure 2.3 for one sample run on the test data set). In addition, the GKHM algorithm also shows the same local behaviour observed in Figure 2.4, and is expected to profit from restricting operations to local updates.

2.2.3 The LBG-U Algorithm

Given that the basic LBG algorithm may become trapped in a local minimum of the distortion error, it is useful to improve a collection of LBG centers. In an approach for improving the basic LBG algorithm, Fritzke [26] suggested to identify cluster centers that contribute only little to the decrease of the distortion error. A *utility* measure $\mathcal{U}$ is employed to quantify this contribution. The unit with minimal utility is relocated to a position in the vicinity of the unit with maximal contribution to the distortion error, after the basic LBG algorithm has coverged. The procedure is stopped after a relocation did not improve the distortion error. The utility $\mathcal{U}(c_i)$ of a center c_i may be quantified as the amount by which the distortion error increases if that center is removed from the set of centers $\mathcal{C}$. Formally the change in the distortion error may be written

$$\mathcal{U}(c_i) = E(\mathcal{X}, \mathcal{C}\backslash c_i) - E(\mathcal{X}, \mathcal{C}).$$

To compute the actual change in distortion error which occurs when a center is removed, it is sufficient to consider the fate of only the points $x^{(i)} \in \mathcal{X}$ which have c_i as their nearest center, i.e., $x^{(i)} = \{x \in \mathcal{X} \mid \mathcal{I}_1(x) = c_i\}$, in other words, the points in the Voronoi set V_i. Each point in V_i is then assigned to its second closest center, $\mathcal{I}_2(x)$, with the change in the distortion error is given by

$$\Delta E(x) = D(c_{\mathcal{I}_1(x)}, x) - D(c_{\mathcal{I}_2(x)}, x).$$

Hence

$$U(c_i) = \sum_{x \in V_i} \Delta E(x).$$

The utility function is used to identify the unit with minimum utility c_{umin}

$$c_{\text{umin}} = \arg\min_{c \in \mathcal{C}} U(c),$$

which is moved into the neighborhood of the center c_{emax} with maximal distortion error

$$c_{\text{emax}} = \arg\max_{c \in C} E(c).$$

In order to assure a useful initialization of the relocated unit c_{umin}, a small offset vector $c_{\text{offset}} = \alpha u$, $\alpha \in \mathbb{R}$, $u \in \mathbb{R}^n$ is added to the center of maximal error c_{emax}, where u is a normed random vector with $||u|| = 1$:

$$c_{\text{umin}} := c_{\text{emax}} + c_{\text{offset}}. \tag{2.8}$$

A useful value for α is the scaled standard deviation of all $x \in V_{\text{emax}}$, given by $\varepsilon\sqrt{E(c_{\text{emax}})/|V_{\text{emax}}|}$, where $0 < \varepsilon \ll 1$, resulting in a new location for c_{umin}

$$c_{\text{umin}} := c_{\text{emax}} + \varepsilon\sqrt{E(c_{\text{emax}})/|V_{\text{emax}}|}u.$$

The removal of the unit with minimal utility, denoted by c_{umin} results in the assignment of $x \in V_{c_{\text{umin}}}$ to $V_{\mathcal{I}_2(x)}$. For the application of the LBG-U Algorithm 2.2 to the test data set see Figure 2.3. We notice in Figure 2.3 that despite the improvement in the distortion error by relocation, there is significant overhead in the LBG-U algorithm, which is obvious from the occurrence of long plateaus during which no significant error reduction can be achieved. This is due to the requirement of applying the LBG algorithm to the new (relocated) set of clusters until convergence. It will be shown that an efficient and useful time for relocation can be identified without unnecessarily prolonging the relocation procedure, by rather assigning c_{umin} and c_{emax} at an intermediate stage. Furthermore, despite the fact that membership changes caused by moving c_{umin} involve only points close to c_{umin} and c_{emax}, consecutive LBG iterations still perform global updates in D with respect to all data points in $\mathcal{X}$. However, since only local membership changes of data points are likely, there is no need in updating a full row of D. Also, sorting should be restricted to cluster centers that are close according to the connectivity matrix,

only these hold potential new closest data points to c_{umin} after relocation. Simulations have also shown the occurrence of oscillations between areas of low utility and high distortion error which also unnecessarily prolong this algorithm and will eventually lead to the termination with a suboptimal distortion error; with the result that the recovery of a previous configuration of cluster center vectors is necessary. Re-assignment of under-utilized centers has also been proposed by Veprek et *al.* [70] as well as Lee *et al.* [42],[2]. The main difference to Fitzke's approach is the introduction of new utility measures, based on additional *ad hoc* parameters to estimate the desired distortion error reduction. In addition all known relocation algorithms also require full convergence of the LBG algorithm to re-assign data points.

Algorithm 2.2 LBG-U($\mathcal{C}, \mathcal{X}$)

Input: Initial (e.g. random) cluster center $\mathcal{C}$, data $\mathcal{X}$
Output: $\mathcal{C}, \mathcal{I}_1, \mathcal{I}_2$

```
Improve ⇐ 1
while Improve do
  C_best ⇐ C
  LBG(C, X)
  if E(X, C) < E(X, C_best) then
    Find c_umin and c_emax
    Move c_umin ≈ c_emax
    Sort for I_1, I_2 {Closest c_i determine Voronoi cell V_i}
  else
    Improve⇐ 0
  end if
end while
```

2.3 The Bi-Directional Hebb Rule

The connectivity matrix $\mathcal{A}$ generated by the CHR is not suitable for establishing a concept of locality for the purpose of identifying neighborhood relations

[2]All articles referring to utility measures and relocation lack the appropriate cross-references.

between centers and data points if the number of centers is less than that which is required to have a dense cluster center distribution. In addition, even with a large number of cluster centers, an unfavorable initial distribution will not resolve enough structural detail to produce dense centers throughout the course of clustering. We observe in Figure 2.2 that the CHR produces connections between clusters that are significantly separated in space. Thus, to reveal distinct regions, or data regimes, we propose to sever the connections between cluster centers generated by the CHR if they are deemed to be separated with respect to a new connectivity measure. This may be accomplished by introducing *bi-directional* competitive Hebbian rule:

Step 1 Form connections according to Hebb rule:

$$A(i,j) = 1 \Leftrightarrow \exists\, x \in V_{ij}, x \in \mathcal{X}.$$

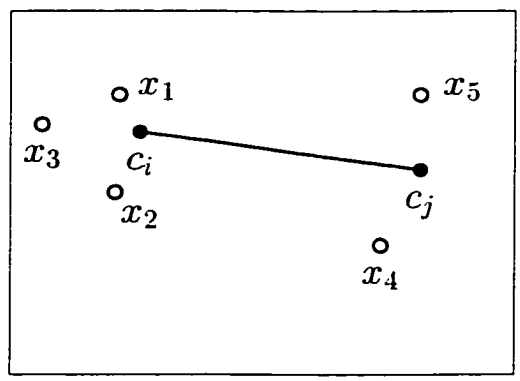
Step 2 Connect only units that have bi-directional interest in connection:
We define the sets

$$\begin{aligned} U_{ij} &= \{x \in \mathcal{X} \mid x \in V_{ij} \cap V_i\} \\ U_{ji} &= \{x \in \mathcal{X} \mid x \in V_{ij} \cap V_j\} \end{aligned}$$

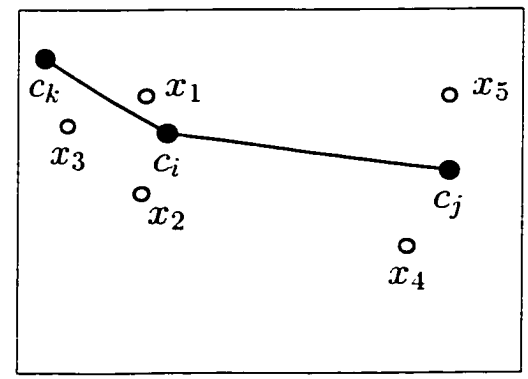
of points $U_{ij} \in V_i$ ($U_{ji} \in V_j$) and that are responsible for connection to c_j (c_i). If one of the sets U_{ij} or U_{ji} is empty, then the connection is concluded to be of only uni-directional interest and therefore should be removed

$$A(i,j) = 1 \Leftrightarrow U_{ij} \neq \{\emptyset\} \vee U_{ji} \neq \{\emptyset\}. \tag{2.9}$$

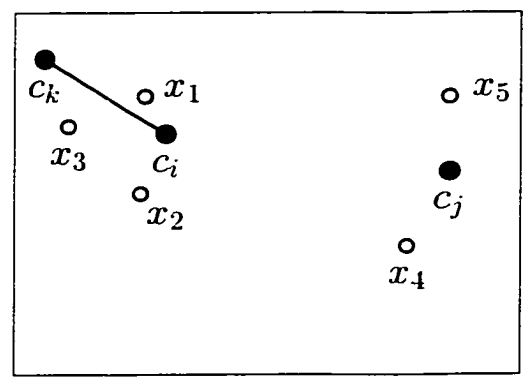
One notes that $V_{ij} = U_{ij} \cup U_{ji}$ and that $V_{ij} = V_{ji}$, but $U_{ij} \neq U_{ji}$. As an example of this procedure consider Figure 2.5. The two cluster centers c_i and c_j in Figure 2.5 (a) are augmented by a third cluster center as shown in Figure 2.5 (b). After the new cluster center c_k is added, the set U_{ij} becomes empty indicating that the connection established by Hebb's rule should be deleted. The result of applying the bi-directional Hebbian rule is shown in Figure 2.5 (c).



(a) Hebb connection between centers c_i and c_j: $V_{ij} = \{x_1, .., x_5\}$, $U_{ij} = \{x_1, x_2, x_3\}$, $U_{ji} = \{x_4, x_5\}$



(b) Hebb connection uni-directional as result of new center c_k: $V_{ij} = \{x_4, x_5\}$, $U_{ij} = \{\emptyset\}$, $U_{ji} = \{x_4, x_5\}$



(c) As a result of $U_{ij} = \{\emptyset\}$, Hebb connection is removed.

Figure 2.5: Contrasting the connections produced by the standard competitive Hebbian rule (see (b)) and the bi-directional Hebbian rule (see (c)) as a consequence of the addition of a new center to the configuration in (a).

2.3.1 Weighted Bi-Directional Hebb Rule

Experimentation with this bi-directional Hebbian rule for creating a set of disjoint triangulation of the data reveals that certain connections rely on only a small number of points that have common nearest and second nearest neighbors. In many instances, it may be deemed that those connections are not representative of a true connection relation between cells. Such connections may be deleted by introducing a weighted bi-directional competitive Hebbian rule.

Step 3 As a measure of the nearness of two Voronoi cells we introduce the weight $w_{ij} \in \mathbb{R}$ defined as

$$w_{ij} = \frac{|U_{ij}|}{|V_{ij}|} \quad 0 \leq w_{ij} \leq 1. \tag{2.10}$$

Now cluster centers are only connected if the bi-directional strengths or weights $w_{ij} \in \mathbb{R}$ or $w_{ji} \in \mathbb{R}$ exceed a certain threshold $s > 0$

$$A(i,j) = 1 \Leftrightarrow (w_{ij} > s) \vee (w_{ji} > s). \tag{2.11}$$

Observe that $w_{ij} + w_{ji} = 1$ and hence a heavy asymmetry between w_{ij} and w_{ji} indicates a significant separation between V_i and V_j. See Algorithm 2.3.

From equation (2.10) we conclude that maximum asymmetry occurs when $w_{ij} = 0$ and $w_{ji} = 1$, therefore if $s > 0$, condition (2.11) includes condition (2.9) as a special case of maximal separation. Optimal balance between two clusters is attained when $w_{ij} = 1/2$ and $w_{ji} = 1/2$. Possible choices for s are

$$s \approx \min_{w_{ij} \neq 0} w_{ij} \quad \text{or} \quad s \approx 1/2. \tag{2.12}$$

In the following we refer to a *structure* $S(c_i)$, associated with a center c_i, as the unit c_i itself and its connected cluster centers

$$S(c_i) = \{c_i\} \cup \{c_m | \, \mathcal{A}(c_i, c_m) = 1\}. \tag{2.13}$$

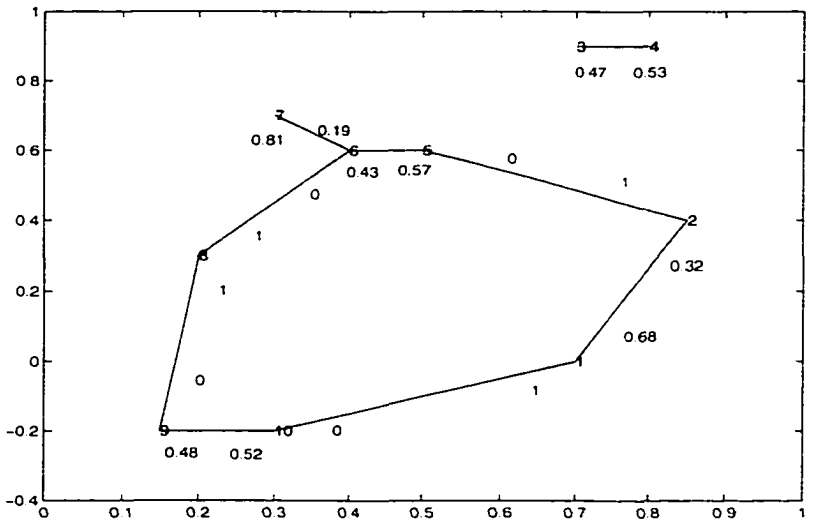


Figure 2.6: Connections established according to competitive Hebbian rule between 10 centers (numbers 1-10) from a Gaussian mixture. The numbers close to one center and shifted slightly to a connecting center indicate w_{ij}, e.g. $w_{12} = 0.68$, $w_{51} = 0$.

Figure 2.6 shows the weights assigned to each connection produced by the CHR. First, all maximal asymmetric connections are removed (Figure 2.7). After keeping connections with weights $w_{ij} > 0.4$ (Figure 2.8) local regions are sufficiently separated. Note that this procedure is also sensitive to the number of local clusters, e.g., in Figure 2.8 cluster 7 is separated from 6, whereas clusters 9 and 10 still form a structure. The reason is that with 3 clusters $7 - 6 - 5$, a higher resolution can be achieved (more data points can be shared by second closest centers) and a breakup into smaller regions is possible. Clusters $9 - 10$ are further away and share all second closest neighbors and therefore still form a structure.

Several ways are possible to further utilize the concept of assigning weights to connections which include the removal of clusters as well as the insertion of new cluster centers in areas of weak or asymmetric connections. This may also help to find algorithms which produce a dense center distributions.

The removal of under-utilized cluster centers and their associated connections has been considered in, e.g., [24, 25]. The elimination of connections that are not

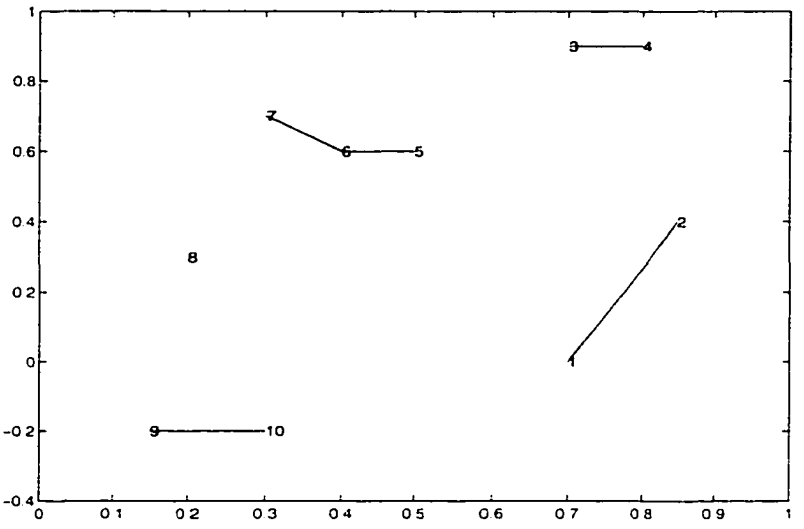


Figure 2.7: Resulting connections after all units of uni-directional interest have been removed (connections of extreme imbalance, $U_{ij} = \{\emptyset\}$).

representative of the topological arrangement of the data (under the assumption of a dense center distribution, discussed in Section 2.1) was implemented in [52]. These algorithms have an on-line architecture in common and require the setting of additional learning and threshold parameters to assess the *age* of a connection. The weighted bi-directional Hebb Rule is formulated to be applied in a batch processing mode. The setting of the threshold parameter s is therefore adjustable after the application of the CHR (see Equation (2.12)) and independent of on-line learning parameters.

Algorithm 2.3 Connectivity($\mathcal{I}_1, \mathcal{I}_2, s$)

Input: $\mathcal{I}_1, \mathcal{I}_2, s$

Output: Modified $\mathcal{A}$ using bi-connectivity

Hebb Rule: $\mathcal{A}(\mathcal{I}_1, \mathcal{I}_2) = 1$

Remove only uni-directional connections

or remove weak connections using threshold: $(w_{ij} > s) \vee (w_{ji} > s)$

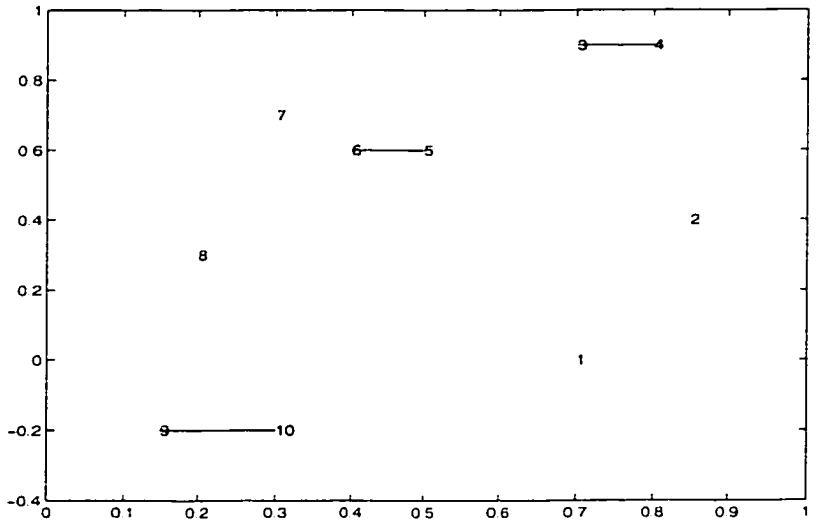


Figure 2.8: Result of removing connections with $w_{ij} < 0.4$ produces additional disconnected regions.

2.4 Exploiting Local Neighborhood Relations

2.4.1 The Local Stage

The computations and updates, e.g., of the distance matrix can be restricted after some initial iterations to local operations based on the following considerations (see Figure 2.4). After some iterations, changes in membership of points only occur at the edge of Voronoi cells (see Figure 2.1) and therefore involve only data points that lie in neighboring cells determined by the connectivity matrix. Therefore, we define T_l to be the iteration where the algorithm switches from a global to a local adaption, defined as:

$$m_k^{(T_l)}(x) \rightarrow m_l^{(T_l+1)}(x), \text{where } l = k \vee A(l,k) = 1 \; \forall \; x \tag{2.14}$$

with the result of membership changes of a point x from $\mathcal{I}_1(x)$ only to $\mathcal{I}_2(x)$ or $\mathcal{I}_2(x)$ only to $\mathcal{I}_1(x)$ for $t > T_l$. The restriction $A(l,k) = 1$ in (2.14) can be relaxed if no bi-directional Hebb rule is implemented and only the sets $\mathcal{I}_1$ and $\mathcal{I}_2$ are updated. Indeed, as Figure 2.4 indicates, after only two iterations all membership changes occur locally and the algorithm is thereafter in a "local stage".

A local distance update based on the previous observations can be designed as follows. In order to account for changes that involve the computation of a cluster center c_i via D, only data points are used that are members of the Voronoi cell V_i and second nearest to c_i or possible candidates to become second nearest, i.e., points that are members of Voronoi cells adjacent to c_i. This ensures that the second order Voronoi cells are updated correctly. The local distance update is summarized in Algorithm 2.4. This computation involves not a full row (of size P) distance matrix update, as it was necessary for the LBG algorithm, but only of order $|\mathcal{X}_l|$, where

$$\mathcal{X}_l = \{x \mid x \in V_i \vee x \in V_j, \text{ iff } A(i,j) = 1\}.$$

Algorithm 2.4 LocalDistanceUpdate($c_i, \mathcal{A}$) for $T > T_l$

Input: $c_i \in \mathcal{C}$, $\mathcal{A}$
Output: $D(x, \mathcal{X}_l)$
Find set $\mathcal{X}_l = \{x \mid x \in V_i \vee x \in V_j, \text{ iff } A(i,j) = 1\}$
Update $D(c_i, \mathcal{X}_l)$

2.4.2 Nearest Neighbors Search

As a consequence of defining iterations where only local changes occur and having appropriate neighborhoods defined via the adjacency matrix we are able to restrict the exhaustive search used in LBG and LBG-U to a local nearest neighbor search. Of interest are the set of indices $\mathcal{I}_1$ and $\mathcal{I}_2$, which can be found for each center c_i among those which are adjacent, i.e., $\mathcal{A}(i,k) = 1 \ \forall k$. Furthermore, applying the bi-directional Hebbian rule reduces the number of connections and therefore restricts the nearest neighbor search to structures obtained by Algorithm 2.4.

2.5 The Local LBG Algorithm (LLBG)

The algorithm we propose has as its core the original LBG algorithm and takes local operations and restricted connections into account. Therefore we shall call it Local-LBG or LLBG. As stated in the previous sections, the algorithm switches from a global state of operation to a local one. Two major efficiency improvements are the essential innovations that recommend the LLBG over the LBG algorithm.

First, we compared the distance matrix update necessary in the LLBG to the LBG algorithm. For this purpose we define the following fractional distance update measure, as the ratio of the number of data points contained in the local data set $\mathcal{X}_l$ over all data points $\mathcal{X}$. Only $|\mathcal{X}_l|$ points will be involved in a row update in the LLBG algorithm:

$$\mathcal{F} = \frac{< \overline{|\mathcal{X}_l|} >}{|\mathcal{X}|},$$

where the mean ($\overline{|\mathcal{X}_l|}$) is taken over all LLBG runs for *one* initial cluster center distributions and averaged ($< \cdot >$) over 100 different runs with new initial center distributions on the test data set. The fraction for the LBG algorithm is $\mathcal{F}$=1. indicating global distance matrix updates. Figure 2.9 shows this fraction for the LLBG algorithm as a function of the number of centers. The lowest value and the most efficient performance occur for approximately 40% of the LBG distance updates at 10 centers. The performance on a test data set of 5000 points with 10 centers is summarized in Table 2.1.

Secondly, we introduce a new termination criteria for the LBG algorithm when the bi-directional Hebb rule is employed. If the number of points that changed membership is less than the number of points responsible for the weakest connection, $|U_{\text{weak}}|$, the LLBG algorithm terminates, that is, if

$$|\{x|\mathcal{I}_1(x) \neq \text{old}\mathcal{I}_1(x)\}| < |U_{\text{weak}}|.$$

Algorithm	Distortion Error	iterations	Time/Run[s]
LBG	6.3884e4	34	14.89
LLBG	6.3901e4	11	3.21

Table 2.1: Simulation results on the sample data set of size 5000 using 10 centers, averaged over 100 runs.

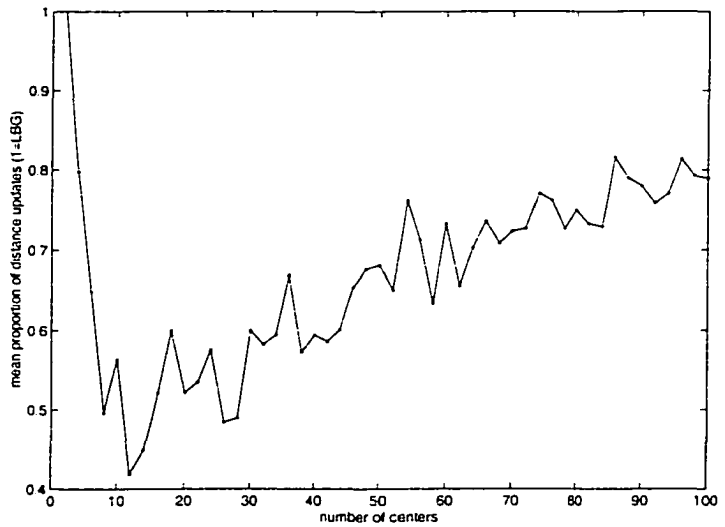


Figure 2.9: This figure compares the necessary number of patterns used in updates of the distance matrix as a ratio. The LBG algorithm in its global implementation uses all patterns in each iteration, whereas the Local LBG only uses patterns in the vicinity of a center that was moved, $|\mathcal{X}_l|$. The saturation effect that occurs as the number of centers increase is a consequence of an increase in the number of neighbors of each center, and is a characteristic of the particular sample data set used.

The performance improvement with respect to increasing number of data points (from the same mixture distribution) for 10 centers is shown in Figure 2.10, together with a polynomial interpolation of order two. These results suggest that an efficiency improvements was achieved due to the locality of the algorithm and the introduction of the novel termination criterion.

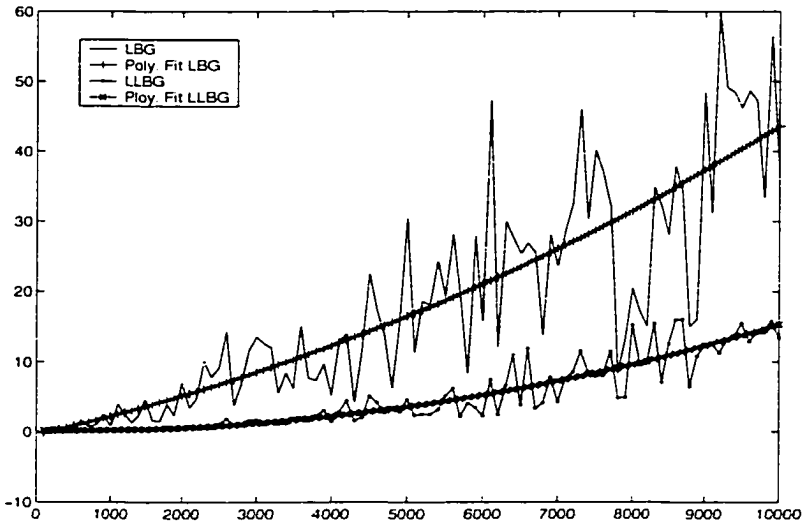


Figure 2.10: Comparison between CPU time needed to compete one LBG or LLBG run until termination.

Algorithm 2.5 LLBG($\mathcal{C}, \mathcal{X}$)

Input: Initial (e.g. random) cluster clusters $\mathcal{C}$, data $\mathcal{D}$
Output: $\mathcal{C}, \mathcal{A}, \mathcal{I}_1, \mathcal{I}_2$

```
Initialize oldC ≠ C
GLOBAL ⇐ 1
while C ≠ oldC {or alternative termination} do
  if GLOBAL then
    Update D(c_i, X)
  else {LOCAL (T > T_l)}
    LocalDistanceUpdate(c_i, A) {Algorithm 2.4}
  end if
  A-restricted search for I_1 and I_2
  A =Connectivity(I_1, I_2) {Algorithm 2.3}
  if GLOBAL (T < T_l) then
    Check for GLOBAL or LOCAL
  end if
  Update C
end while
```

2.6 Utility Based LLBG (LLBG-U)

LBG and LBG-U both have a major inefficiency due to the global nature of the algorithms. We therefore propose a rule based relocation with takes prior observations about the integration[3] and the relocation of cluster centers into account. As stated above, the relocation and integration should involve only local updates to D. Also, there is no need to let the LBG algorithm converge, but rather perform relocations after the algorithm becomes local, i.e., for $T > T_l$.

2.6.1 Rule Based Relocations

The following list of rules propose a new concept of changing the locations of c_{umin}, based on the utility measure proposed by Fritzke [26].

R1 Integration: A relocated unit cannot be identified as c_{umin} until it has been integrated, i.e, membership changes for the relocated unit satisfy the locality condition (2.14).

R2 One cycle: If a unit leaves a connection structure (see (2.13)), one relocation into the same structure is permitted. Thereafter the structure freezes and no further relocations are permitted. This rule prohibits undesirable oscillations between regions of high distortion errors.

R3 Local integration: Involve only local data in integration or relocation procedures. Update only $D(c_{\text{emax}}, \mathcal{X}_l)$ and $D(c_{\text{umin}}, \mathcal{X}_l)$.

R4 No neighborhood help: If c_{umin} and c_{emax} are connected ($\mathcal{A}(c_{\text{umin}}, c_{\text{emax}}) = 1$) no relocation is performed.

---

[3]The term *integration* refers to the operations associated with updating the distance matrix and sorting for $\mathcal{I}_1$ and $\mathcal{I}_2$, if a center configuration has been changed due to the relocation of a cluster center.

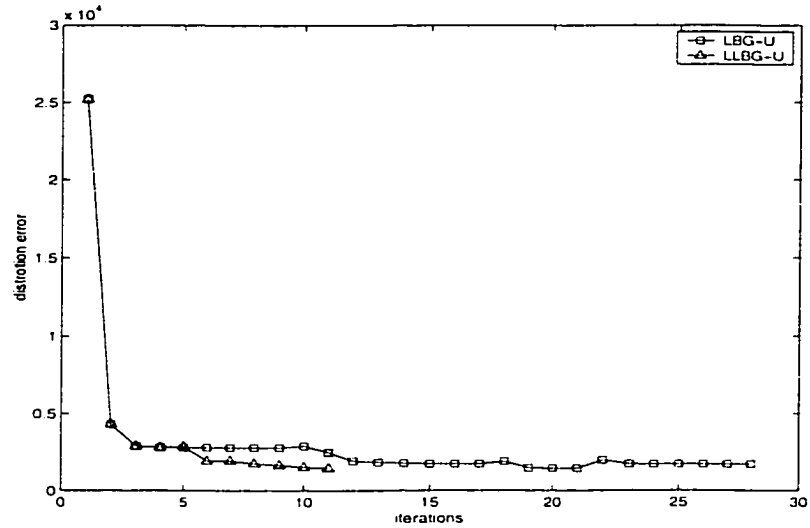


Figure 2.11: One typical run compares the performance of the LBG-U and LLBG-U algorithms on the distortion error.

The LLBG-U has as its core again the local operations that were used to improve the efficiency of the LBG algorithm combined with center relocations using a utility measure. The novel approach here is to perform relocation during one LBG cycle with the restrictions formulated in R1-R4. Figure 2.11 shows one typical run of the algorithms, whereas Figure 2.12 presents an averaged performance comparison over 1000 runs on 1000 test data points.

The comparisons between LBG-U and LLBG-U show a faster convergence to a lower distortion error due to the fact that relocations are possible during one LLBG cycle (see Figure 2.12 for a single run and Figure 2.12 for an average of 1000 runs); the mean number of iterations until convergence are 11 for LLBG-U and 37 for LBG-U, with approximately the same expense in each iteration, resulting in a considerable speedup of the LLBG-U employing the set of rules introduced above.

Algorithm 2.6 LLBG-U($\mathcal{C}, \mathcal{X}$)

Input: (e.g random) $\mathcal{C}$, data $\mathcal{X}$

Output:

```
Initialize old𝒞 ≠ 𝒞
GLOBAL ⇐ 1
RELOCATE ⇐ 1
while RELOCATE do
  while 𝒞 ≠ old𝒞 do
    if GLOBAL then
      Update D(c_i, 𝒳)
    else {LOCAL}
      LocalDistanceUpdate(c_i, 𝒜) {Algorithm 2.4}
    end if
    𝒜-restricted search for ℐ_1 and ℐ_2
    𝒜=Connectivity(ℐ_1, ℐ_2) {Algorithm 2.3}
    Check GLOBAL or LOCAL
    Determine c_umin and c_emax
    if R1, R2, R3 and R4 then
      c_umin ≈ c_emax
      Update neighbor and frozen structures
    else {No relocation possible}
      Update neighbor and frozen structures
    end if
    if All structures and centers exhausted then
      RELOCATE⇐ 0
    end if
  end while
end while
```

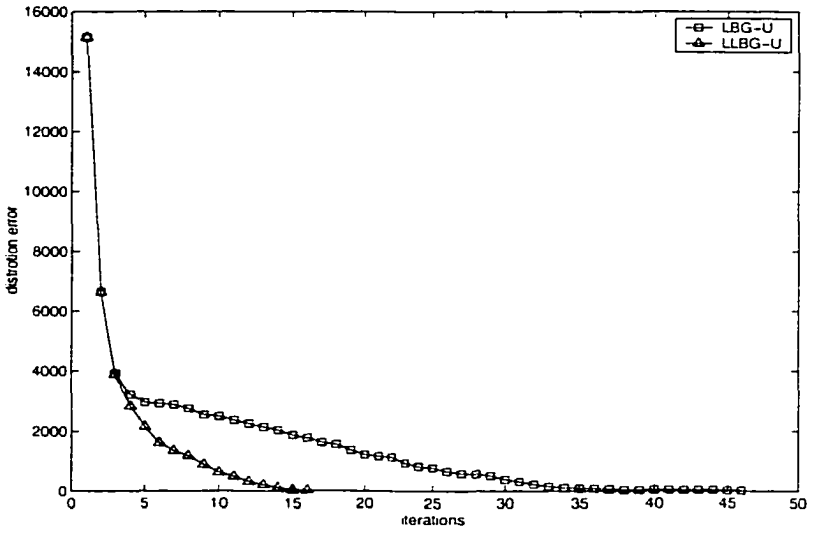


Figure 2.12: The averaged performance of LBG-U and LLBG-U with 10 centers over 1000 runs on the test data set of size 1000.

2.7 The Growing LLBG-U Algorithm

The same considerations about neighborhood adaptations made in the discussion of LBG or LBG-U apply to the scenario where we consider the growing LBG algorithm [2]. Growing refers to the process of adding cluster centers to the set C in order to be more adaptive to the data. The LLBG-U algorithm terminates with a center distribution where no significant decrease in the distortion error can be achieved by moving a center. Therefore it is necessary to insert a new center in the neighborhood of c_{emax}. This new center is integrated in the cluster center set the same way as c_{umin} is. Again, the growing LLBG-U (G-LLBG-U) results in significant decrease in cpu-time compared to the growing LBG, where convergence is required before we insert a new center in the vicinity of c_{emax} (see Figure 2.14). Again, due to the rule based relocation and early integration of new clusters the growing version of LLBG-U shows faster convergence and earlier termination as can be seen in Figure 2.13.

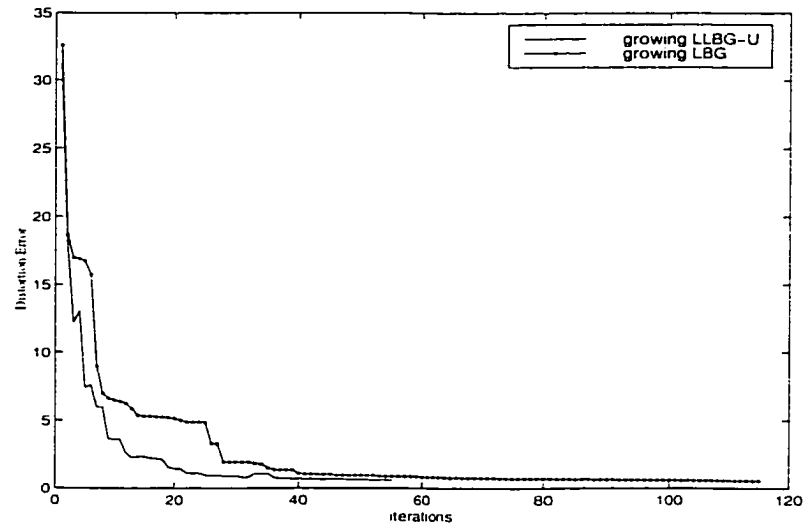


Figure 2.13: Comparison of the two growing versions of LBG and LLUBG-U on the distortion error.

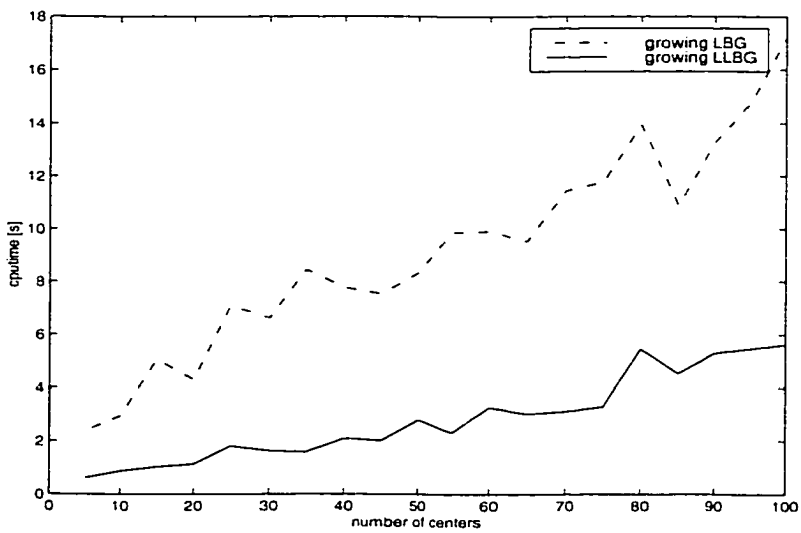


Figure 2.14: As the number of centers increases the CPU time was recorded as a function of centers for the growing versions of LLBG and LBG.

2.8 An Application: Structure in High-Dimensional Data Sets

Massive data sets generated by potentially nonstationary physical processes are often characterized by the fact that they occupy disjoint subsets in high dimensional space. This coherence, or organized structure in the data may arise, for example, in dynamical systems as a result of shifting (unstable) steady states. Industrial processes often have distinct operating regimes, each characterized by occupying a specific region of the measured state space of the system. The identification of the operating regime based on collected data may be formulated as a subset identification problem. In this application we address the issue of constructing uncoupled Delauney Triangulation via the connectivity matrix generated by a bi-directional Hebbian rule. The resulting data structure provides a quantification of the number of different *regimes* present in the data and their state space location. Such a characterization provides a significant data reduction as well as a basis for analyzing the data generated by a wide range of physical phenomena.

2.9 An Example: Analyzing an Industrial Process

In this example we consider an industrial process[4] represented by five time series collected for 2560 time units. The process is known to have many (unlabled) operating regimes and the task is to notify the field technician as to which of the operating regimes is currently in effect.

The data matrix of size 5×2560 has full row rank so any projections of the data will necessarily lose information. Thus all of the calculations are performed in the original unscaled coordinate system.[5] Three of the five time-series are plotted

---

[4]This data was provided by Honeywell Inc.

[5]Scaling data in order to emphasize the disconnected regions is a complicated task. In this application it was found that little improvement in the results was obtained by applying the usual scaling techniques.

in Figure 2.15; it is seen that there are indeed several operating modes characterized by clumps or clusters. The distinct regions, or modes, are identified by the connections generated by the bi-directional Hebbian rule as shown in Figure 2.15. The LLBG algorithm with bi-directional Hebb rule was used for clustering with 10 centers and a threshold $s = \min_{w_{ij} \neq 0} w_{ij}$. The plots show projections on $\mathbb{R}^3$ of the disconnected regions as opposed to the actual disjoint regions residing in $\mathbb{R}^5$.

Finally, the actual five time-series are shown in Figure 2.16 using separators for the disjoint regions and numbers to identify the associated cluster centers. In practice, it is envisioned that a systems expert will assist in the initial identification of operating modes and that, once calibrated, these graphs will provide robust diagnosis tools for complex industrial processes. Furthermore, the identified modes of operation can be generalized to future states of the process using the same equipment; once a process has visited all valid states, a classification of the current state is possible and a failure of the equipment can be detected.

2.10 Discussion and Summary

A discussion of LBG based clustering algorithms revealed that the concept of locality defined by closest and second closest cluster centers can be used to increase the efficiency of vector quantization procedures. Furthermore the concept of identifying local data regions via connectivity using CHR was exploited for the computations of a utility measure to find under-utilized cluster centers. A bi-directional Hebbian algorithm is presented that is capable of quantifying disjoint regions of data in a high-dimensional phase space. The algorithm is shown to disconnect regions that would be connected in an induced Delauney triangulation produced by the uni-directional Hebbian rule. The resulting characterization of the data has proven useful for analyzing industrial processes. Future work will include the development of alarm mechanisms for the diagnosis of faults in industrial processes. We are also continuing our investigation of fast algorithms for

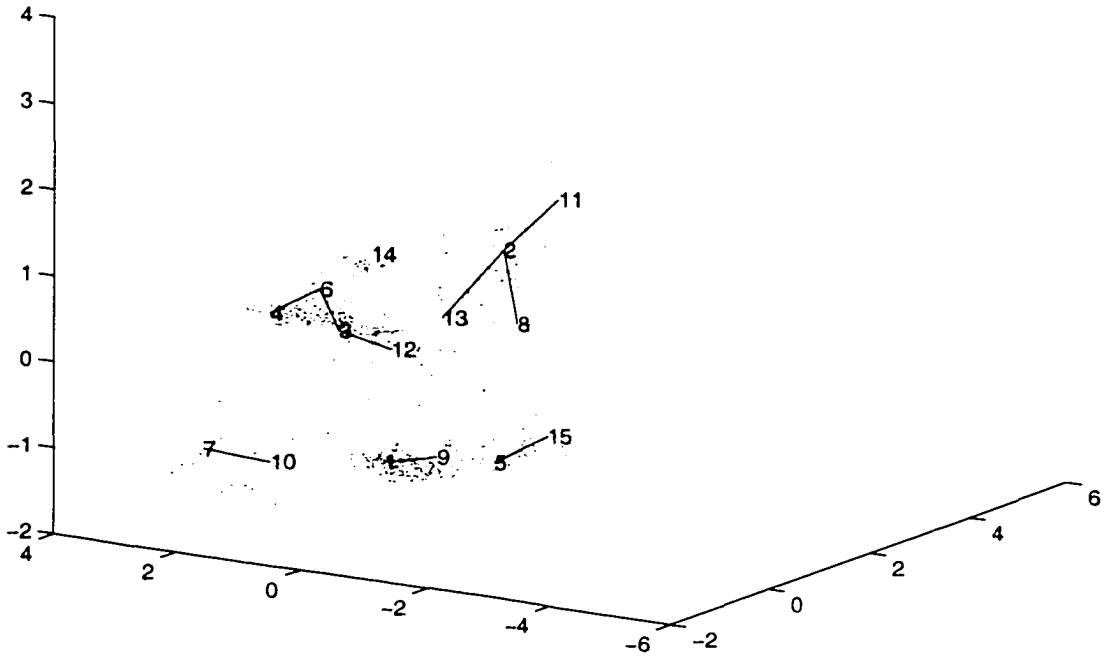


Figure 2.15: The disconnected regions produced by the weighted bi-directional Hebbian rule are shown and correspond to different operating regions. Numbers $1-15$ indicate the position of cluster centers, lines connecting only centers within the same mode of operation. E.g. clusters $13, 18, 2, 11$ correspond to one operating regime.

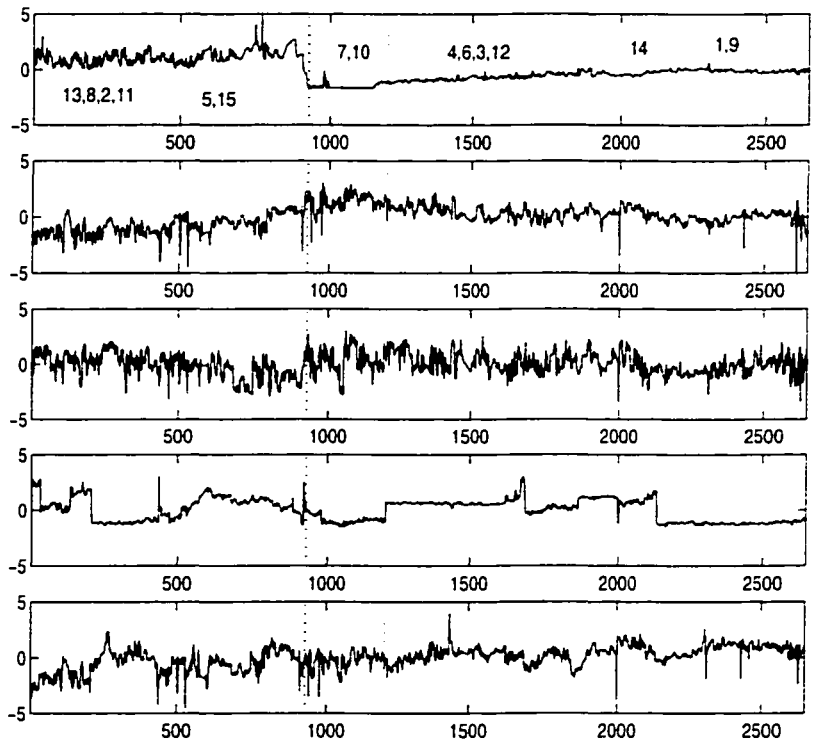


Figure 2.16: The original five industrial process time series are shown using separators (dashed line) to distinguish distinct operating modes identified from disconnected regions. The numbers in the top figure are the associated clusters from Figure 2.15. The first transition is observed when the trajectory moves away from the disjoint region defined by clusters $13, 8, 2, 11$. The new regime is identified by the connected clusters $5, 15$. The process evolves through a sequence of six regimes

clustering the data that permit a growing number of centers based explicitly on the connectivity structure. In addition we seek to develop a principled approach to detect to number of clusters bases on statistical validation. Related scientific applications also include the identification of states in chaotic dynamical systems, structural image analysis and processing of data to be classified.

Chapter 3

RADIAL BASIS FUNCTION NETWORKS BASED ON AUTOCORRELATION FEEDBACK RESOURCE ALLOCATION

A model validation test based on simple linear autocorrelations is proposed as an objective method to determine the optimal number of units in the hidden layer of a radial basis function network. The data to be fit is assumed to consist of a signal with additive, independent identical distributed (iid) noise. A novel stopping criteria is introduced based on the statistics of the residuals rather than on *ad hoc* parameters. Consequently, this network is shown to neither overfit nor underfit the data.

3.1 Introduction

Radial basis function (RBF) networks have proven to be attractive for solving a wide-range of approximation and classification problems [9, 39]. Their appeal is due, at least in part, to the mathematical tractability of their basic architecture which consists of one input layer, one nonlinear hidden layer and one linear output layer. In general, the size of the input and output layers is specified by the dimension of the data in the problem under investigation, whereas the dimension of the hidden-layer and its nonlinear parameters (centers and widths) are unknown *a priori* and must be determined by the training process. There are also many options for computing the training parameters: the location of the basis

functions may be determined either by clustering algorithms [57] or optimization methods [9]; the widths of the basis functions may be chosen heuristically [32] or via nonlinear optimization techniques; the remaining weight parameters may be computed via the singular value decomposition [11] (or other direct methods for solving overdetermined linear systems) as well as iteratively via descent methods [39]. It is also possible to effectively blend nonlinear and quadratic optimization problems [55, 48]. In addition, the mathematical structure of the RBF learning problem affords a particularly efficient implementation of forward-selection methods for model selection of linear regression models like orthogonal least squares (OLS) [16]. In combination with generalized cross-validation and regularization, RBFs offer a powerful tool to address the bias-variance problem [63].

Adaptive methods for function approximation, such as radial basis functions and their cousins, feedforward sigmoidal neural networks, may also be based on *growing*,[1] or resource allocating network (RAN) algorithms as proposed by Platt [64]. Growing architectures are applicable to batch data sets, but they are especially attractive for data arriving in streams. Such a growing algorithm is especially suited to nonstationary data since the network parameters themselves may be nonstationary over time. In particular, such constructive network topologies offer an attractive alternative to fixed, i.e., non-growing, topologies as they may be adjusted such that the complexity of the model matches the complexity of the data, as demonstrated, e.g., in an application to reinforcement learning [4].

During the course of training via a growing algorithm, new data is presented to the network and a decision must be made concerning the acceptibility of the model. If the new data is deemed sufficiently novel (typically via one or more *ad*

[1]Here we understand that growing networks may grow larger or even become smaller if one or more units have become obsolete.

hoc parameters) then a new descriptive basis function, or unit, is added; otherwise the existing model is adapted to the new data using a least mean square algorithm (LMS) [71]. Many alterations and improvements of this basic procedure have been introduced. The replacement of the LMS with the extended Kalman Filter resulted in more compact network architecture and faster convergence (RANEKF) [38]. The identification of hidden units with little utility in subsequent stages of the algorithm and their removal, as well as the addition of a windowed error-threshold lead to the M-RAN network [72, 73]. Statistical criteria have also been proposed to overcome the shortcomings of the *ad hoc* novelty criteria but these require detailed knowledge of the statistics including an estimate of the covariance matrix for the noise process from the Kalman Filter approach. In [69] a pointwise error test was proposed and extended to a whiteness test in [54, 53].

Studies that evaluate these methods to assess the role of the numerous *ad hoc* parameters leave the reader with the impression that modeling with RANs is a combination of science and art [47, 61, 54]. The performance of RANs appears to be especially problematic for nonstationary data where recursive training (cycling repeatedly through the training data) is not possible. The proposed training parameters typically must be significantly adjusted by the user to avoid overfitting and retain good generalization, a procedure that requires extensive knowledge of the data and the results of which become invalid as the data changes. In other words, these RANs are essentially only useful for addressing the model order problem as this user-selected set of parameters essentially determines the nature of the resulting neural approximation and requires prior information from trial-and-error runs which disqualifies them from being truly on-line methods.

This paper proposes a RAN training algorithm that eliminates the need for extensive set of *ad hoc* parameters of current state-of-the-art RAN methods. The premise of the algorithm is that a good RBF model will be obtained if the residuals

have the appropriate statistics. For the purposes of this paper we will assume that the noise is additive and iid. The RBF algorithm then consists of a growing network that tests the autocorrelation of the residuals at each step. The stopping criterion now consists of determining whether the residuals are deemed to be iid. If this is not the case, then the order of the model is increased and the location of the new basis function is at the point of a maximum in the autocorrelation function. This evaluation of the autocorrelation function and the resulting center selection is a form of feedback. Hence, we will refer to the method presented in this paper as the autocorrelation feedback RAN RBF (ACF-RAN). It is proposed as an alternative to RAN networks in a batch environment.

3.2 The RBF Resource Allocating Network (RAN)

3.2.1 Radial Basis Functions

The central assumption in using an RBF neural network for data approximation is that the underlying process can be modeled using a three-layer network that implements a mapping $\hat{y} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ according to

$$\hat{y}(x) = w_0 + \sum_{k=1}^{M} w_k \phi(\|x - c_k\|) = w_0 + \sum_{k=1}^{M} o_k(x), \tag{3.1}$$

where the kth response function $\phi(x)$ is centered at c_k and has associated weights $w_k \in \mathbb{R}^m$; the weight w_0 is a constant bias term. The terms $o_k(x)$ are the activations of the hidden layer; the norm $||.||$ is generally taken to be the Euclidian norm. If the domain of the RBF is a compact subset of $\mathbb{R}^n$, then every continuous real valued function may be approximated uniformly by linear combinations of RBFs with centers in the domain; for a proof see the appendix by Brown in [65].

A number of functions, with both local and global support, qualify as radial basis functions including Gaussians, cubics, thin plate splines and multiquadrics [65]. In this paper we restrict our attention to the local Gaussian response function

$$\phi(r) = \exp(-r^2).$$

To provide additional structure to the expansion, we assume that the receptive field of the Gaussian (i.e., domain values for which its range is nonnegligible) is shaped via the width parameters σ_{jk}. This is equivalent to employing a weighted Euclidean norm. Thus

$$\hat{y}(x) = w_0 + \sum_{k=1}^{M} w_k \exp(-\sum_{j=1}^{n} \frac{(x_j - c_{jk})^2}{\sigma_{jk}^2}),$$

where we now focus on the case where the image of the RBF has dimension $m = 1$. The number of basis functions M and the constant term determines the *model order* K, i.e., $K = M + 1$ (if no bias term is included, $K = M$).

3.2.2 RAN Algorithms

The batch training mode of a RAN consists of sequentially adapting the number of radial basis functions until it has been determined that a stable configuration has been reached. In a truly on-line mode the adaptation of the RBF is generally a continuous process as the nature of the data may change, e.g., due to nonstationarity. For both modes initially, $K = 1$, i.e, the algorithm begins with no RBF units having been allocated. At the kth iteration the network is evaluated to determine whether a unit should be added, a unit removed or the number of units should remain constant. The allocation of the new unit $\phi(\|x - c_k\|)$ consists of locating the new center c_k followed by adapting the entire set of augmented weights as well as the new width vector σ_{jk} and center vector c_{jk} parameters. If a new unit is not added, i.e., a unit is either deleted or the order of the model remains constant, then only the weights are adapted to fit the new data. In our implementation the weights are found by solving a least squares problem while the widths and centers are adapted using BFGS, a nonlinear optimization algorithm [49].

In general, RAN methods employ two measures of model performance, viz., the approximation error on the training set $\mathcal{X} = \{(x_1, y_1), .., (x_P, y_P)\}$ as well as

the approximation error on a test set $\mathcal{X}_t = \{(x_1, y_1), ..., (x_{P_t}, y_{P_t})\}$. The error on the test set is generally referred to as the generalization error. As basis functions are allocated, the training error for a noisy data set decreases monotonically. The generalization error, however, will at first decrease and then increase, an indication that the data is being overfit. The common goal of RAN networks is to stop training at the onset of overfitting.

These approximation errors for RANs are typically taken to be root mean square errors (RMSE), i.e.,

$$RMSE = \sqrt{\frac{1}{P}\sum_{i=1}^{P} e_i^2}.$$

Our approach for determining the appropriate order of the RMF model is based on a statistical analysis of the set of residuals at each iteration. For a single observation the model residual is

$$e_n = \hat{y}(x_n) - y_n.$$

We define $R^{(K)}$ to be the set of training residuals for a model of order K, i.e.,

$$R^{(K)} = \{e_n : x_n \in \mathcal{X}, \text{model order K}\}.$$

Note that for a model of order zero these residuals are just the target data.

3.2.3 The M-RAN Method

Here we present the M-RAN method, a state-of-the-art RAN, for the purposes of later comparing this method with that proposed here. For further details see, for example, [72]. M-RAN is a sequential learning approach, utilizing a set of user-supplied parameters $\mathcal{U} = \{\epsilon_{max}, \gamma, \epsilon_{min}, M, e_{min}, e'_{min}, \delta, \kappa\}$. For each pattern (x_n, y_n) presented to the network, the output $\hat{y}_n = \hat{y}(x_n)$ is computed, followed by

a set of errors, where $M > 0$ defines the size of a sliding window,

$$\epsilon_n = \max\{\epsilon_{max}\gamma^n, \epsilon_{min}\}, \quad \text{where } 0 < \gamma < 1 \text{ is a decay constant} \tag{3.2}$$

$$e_n = y_n - \hat{y}_n, \tag{3.3}$$

$$e_{rmse} = \sqrt{\frac{\sum_{i=n-(M-1)}^{n}[y_i - \hat{y}_i]^2}{M}}, \tag{3.4}$$

necessary for the evaluation of the *growth* criteria

$$|e_n| > e_{min} \quad \text{and} \tag{3.5}$$

$$||x_n - c_{nr}|| > \epsilon_n \quad \text{and} \tag{3.6}$$

$$e_{rmse} > e'_{min}, \tag{3.7}$$

where c_{nr} is the center closest to x_n. If all *growth* criteria (3.5)-(3.7) are satisfied, a new unit is allocated with the following new parameters:

$$w_{K+1} = e_n, \tag{3.8}$$

$$c_{K+1} = x_n, \tag{3.9}$$

$$\sigma_{K+1} = \kappa||x_n - c_{nr}||. \tag{3.10}$$

If one of these criteria is not satisfied, then it is deemed that the presented data point is not *novel* and the RBF weights only are adapted with the extended Kalman filter (EKF). The novelty criteria formulated as error thresholds have the following interpretation. e_{min} controls the desired approximation accuracy of the network, ϵ_n represents the scale of resolution in the input space and decays exponentially, controlling the interdistance between new centers. This leads to a center allocation criterion which allows fewer basis functions with large widths initially, while for larger n, more observations become available and smaller widths (controlled by κ in (3.10)) are possible, fine tuning the approximation. This step is followed by the

computation of the activations $o_k^n = o_k(x_n)$ and a normalized utility measure for each unit:

$$r_k^n = \left| \frac{o_k^n}{\max(o_k^n)} \right|.$$

If $r_k^n < \delta$ for M consecutive observations, the kth unit is pruned and the dimensions of the EKF are adjusted accordingly. This pruning criterion removes RBF units which make only an insignificant contribution to the overall network for M training iterations. The threshold on the windowed error e_{rmse} (3.4) ensures the smooth transition between growing and pruning the network.

In all, the set $\mathcal{U}$ consisting of eight parameters have to be set a priori or by trail-and-error. Small perturbations can result in highly variable network sizes with insignificant change in generalization error. Also, as mentioned previously, the M-RAN algorithm is essentially a batch method and does not permit true on-line training as the data must be cycled through several times during the course of training. In addition, experiments have shown that the best values of the *ad hoc* parameters (consequently the network performance) depends highly on the order in which the data is presented. The addition of noise also has a dramatic negative impact on the performance of the M-RAN with a fixed set of parameters; this is illustrated numerically in Section 4.

3.3 Autocorrelation Feedback

Consider the convergence of a RAN when applied to observations with additive iid noise. At the outset of the process of building an RBF model via a growing algorithm, one does not expect the residuals $R^{(K)}$ to be iid. Indeed, they will typically be highly autocorrelated; it is only at the point that the data is accurately modeled, but not overfit, that the residuals appear iid. This observation forms the basis of a parameter-free stopping criterion: the order of the model is correct when the residuals first become iid, in other words, the onset of overtraining is identified

with the recovery of iid residuals. In what follows, we describe how this stopping criterion may be implemented by calculating the autocorrelation function of the residuals.

3.3.1 The Autocorrelation Test

The sample autocorrelation function (ACF) for a set of residuals $e_1, e_2, ..., e_n$ with mean $\bar{e}$ and lag h is defined as

$$\rho(h) = \frac{\gamma(h)}{\gamma(0)},$$

where $-n < h < n$,

$$\gamma(h) = 1/n \sum_{i=1}^{n-|h|} \alpha(h, e_i),$$

and

$$\alpha(h, e_i) = (e_{i+|h|} - \bar{e})(e_{i-|h|} - \bar{e}).$$

For fixed $h = h^*$ the quantity $\alpha(h^*, e_i)$ is the contribution to the autocorrelation by the ith data point; hence, we refer to this pointwise measure as the sample autocorrelation contribution (ACC). For a multidimensional extention of correlations see [8].

In general, if noise is iid with finite variance, then, for $h > 0$, the sample ACF $\rho(h)$, $h > 0$, is approximately iid. Additionally, this distribution is normal with mean zero and variance $1/n$ where n is assumed to be large [10]. Hence, for iid noise, approximately 95% of the values of $\rho(h)$ should satisfy

$$\frac{-1.96}{\sqrt{n}} < \rho(h) < \frac{1.96}{\sqrt{n}}. \tag{3.11}$$

Thus, if our maximum lag value is h, then $.95h$ of the sample ACF values should satisfy Equation (3.11). If this inequality is not satisfied, i.e., more than $0.05h$ values of the sample ACF fall outside the bounds, we should conclude that the noise is actually not iid.

Now, we can use the above facts concerning the estimated distribution of the sample ACF for our RBF modeling problem, for the case where the observations are known to have added iid noise. An RBF model is deemed to be "correct" if it fits the data. For such a model, the residual consists wholly of noise. If the noise is known to be iid, then the ACF of the residuals should pass the above test. In the next section we develop a RAN algorithm that exploits these ideas to produce a stopping criterion with no *ad hoc* parameters.

3.3.2 The Autocorrelation Feedback Algorithm for Allocating New Units

The resource allocating network algorithm we propose is centered around the iid test for the residuals and the actual procedure for allocating a new unit. The iid test was described above and forms the basis of the stopping criterion. Here we develop the remaining components of the algorithm.

If the residuals at the kth iteration, i.e., $R^{(K)}$, fail the iid test, then the location of the $K+1$st unit must be determined. We propose that this new unit should be inserted at the point where the model residual has maximum contribution to the ACF $\gamma(h)$. The point in the domain x_i^* that is responsible for the maximum contribution to the ACF is found in two steps: First the lag h^* is calculated that maximizes the ACF, i.e.,

$$h^* = \arg\max_{h \neq 0} \gamma(h).$$

Secondly, x_i^* is selected to be the domain point that has maximum positive contribution to $\gamma(h^*)$, viz.,

$$x_i^* = \arg\max_{x_i \in \mathcal{X}} \alpha(h^*, e(x_i)).$$

Allocating a radial basis function centered at this point, i.e.,

$$c_{K+1} = x_i^*,$$

will reduce the value of the ACF where it is needed most.

Once a new center location has been established by the above procedure, a local region $\mathcal{X}_{local}$ about the center is determined. The data in this region is then employed for optimizing the parameters in the RBF. The procedure for finding $\mathcal{X}_{local}$ follows the idea that only points in input space close to x^* contribute significantly to the ACF. To account for noise, outliers and sudden jumps, the ACC is smoothed using a windowing technique in input space and evaluated over points that are closest to x^*. First, let $\mathcal{I}_1$ be the index set of input data sorted according to their distance from x^*, e.g. $\mathcal{I}_1(1)$ is the index of point $x_{\mathcal{I}(1)}$ closest to x^*. Now define the smoothed (windowed) ACC function as

$$WACC_M(n) = \sum_{i=n-(M-1)}^{n} \alpha(h^*, x_{\mathcal{I}(i)}) \qquad M \leq n \leq P, \tag{3.12}$$

where $WACC_M(n)$ reaches a minimum that is valid for M consecutive points identifies n^*. The local data $\mathcal{X}_{local}$ is taken to be the n^* closest points to x^*. $\mathcal{X}_{local}$ is then used in adapting the width and the center of the newly allocated basis function at x^* using a quasi-Newton optimization method (BFGS with approximate line search [49]). The linear weights are determined in combination with BFGS, solving the corresponding LS problem, in manner analogous to that of [48] for multilayer perceptrons or [55] for RBF networks. As a result of this local adaption each newly allocated unit matches and responds to the data in $\mathcal{X}_{local}$ only resulting in a compact and parsimonious network architecture.

3.4 Simulation

To compare the ACF-RAN algorithm with other methods we propose to fit the Hermite Polynomial and the MATLAB "Peaks" function. The Hermite polynomial

$$f(x) = 1.1(1 - x + 2x^2)\exp(-\frac{1}{2}x^2)$$

is a scalar function of a single variable and has been used as a benchmark function in many RBF and RAN studies [44, 62, 63]. The "Peaks" function

$$
\begin{aligned}
x =& \frac{3}{1+\exp 0.02t}\cos t \\
y =& \frac{3}{1+\exp 0.02t}\sin t \\
z =& 3(1-x)^2\exp(-(x^2)-(y+1)^2) - \\
& 10(x/5-x^3-y^5)\exp(-x^2-y^2) - 1/3\exp(-(x+1)^2-y^2)
\end{aligned}
\tag{3.13}
$$

is a scalar function of two variables and is introduced here to investigate the issues associated with multi-dimensional domains, considered by other authors only in the noise-free case. Note that our algorithm requires that the data be presented to the network in a time ordered manner. Thus, the data sets were generated using a parameter t that plays the role of time in a time-series.

3.4.1 The M-RAN Performance

First, we investigated the performance of the M-RAN network with the same set of parameters given in [72], including the EKF settings: $\mathcal{U} = \{\epsilon_{max} = 2.0, \epsilon_{min} = 0.2, \gamma = 0.977, e_{min} = 0.02, \kappa = 0.87, e'_{min} = 0.3, M = 25, \delta = 0.01\}$. A training set consisted of 100 randomly sampled domain points from the interval $[-4, 4]$ plus the associated range points from the Hermite polynomial. First, following [72], the performance was evaluated on a test set of 200 equally spaced data points in $[-4, 4]$. To examine the claim in [72] that the above M-RAN training parameters are robust to additive noise applied a second, more stringent test was applied. Now the range data consists of the Hermite polynomial with added Gaussian noise of mean zero and standard deviation 0.2.

In the noise-free case, the M-RAN algorithm allocated 8 units (with a low RMSE of 0.0095) before the stopping criterion was satisfied. However, for the

noisy Hermite data, the resulting M-RAN network algorithm with the same parameters as in the noise-free case allocated 20 units. This result suggests that the data is being overfit and an examination of Figue 3.1 supports this conclusion[2]. Thus, it appears that the *ad hoc* parameters are indeed sensitive to additive noise. Furthermore, the residuals produced by the M-RAN model of the noisy Hermite data are autocorrelated as evidenced by the ACF shown in Figure 3.2. By contrast, the ACF-RAN algorithm will be seen to terminate unit allocation with uncorrelated residuals.

Note that our algorithm is based on computing autocorrelations of time ordered data and that the training data is assumed to be sequential. From this perspective, we view the Hermite polynomial data as being generated by a smooth, if noisy, process. Thus, there are inherent differences in the manner in which these algorithms "see" the data. One could attempt to apply the M-RAN architecture to time ordered data, however, that would require that an entirely new set $\mathcal{U}$ of parameters be established by trial and error—an exercise that our algorithm was designed to avoid. And, as always, a failure to obtain good results for the M-RAN network might always be attributed to not having determined the optimal tuning.

3.4.2 The ACF-RAN Performance

The Hermite Polynomial

For the ACF-RAN, 100 equally spaced points in the interval $[-4, 4]$ were used for training, with normal noise with standard deviation of 0.2 added. The test set is the same as used for the M-RAN example. Figure 3.3(top) shows the ACC computed from the ACF of $R^{(0)}$ (no bias term included) shown in Figure 3.4(top).

---

[2]Note, M-RAN does not include a direct calculation of the cross-validation error, which might avoid such overfitting.

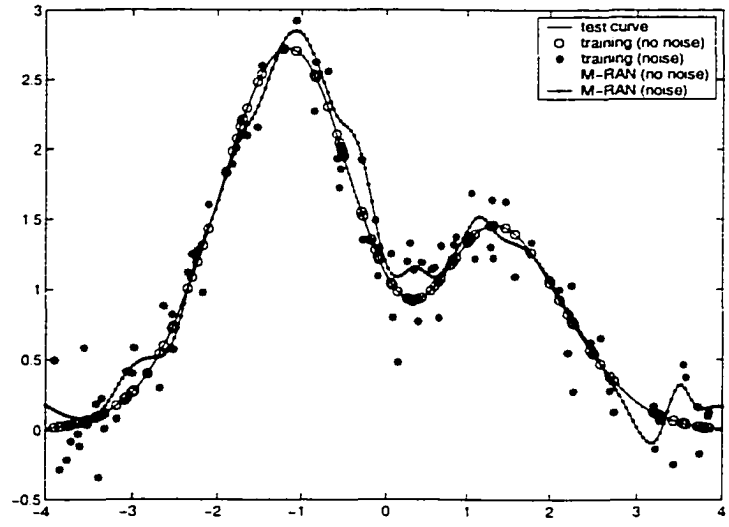


Figure 3.1: The M-RAN fit using the noise-fee training data gives an almost perfect reconstruction with a final RMSE of 0.0095. Leaving the parameters unchanged for the noisy training data results in a poor fit.

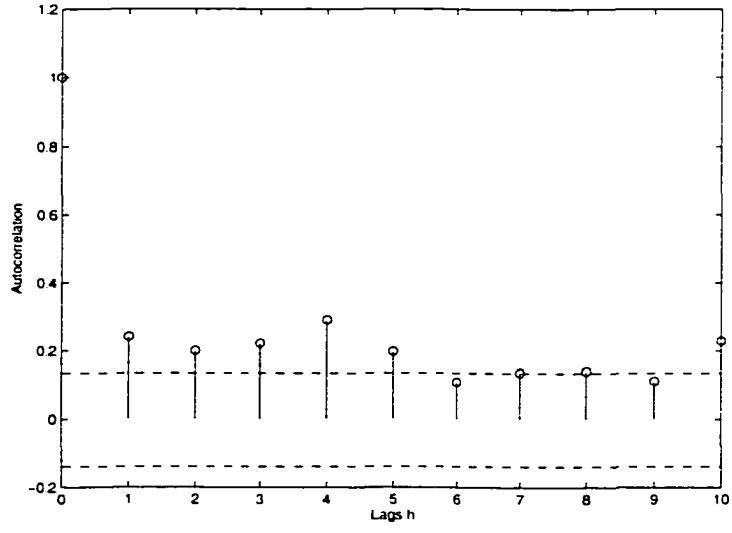


Figure 3.2: The autocorrelation function for the noisy test data shows that the M-RAN fails to reproduce iid residuals for the Hermite polynomial.

After only 2 units are allocated all autocorrrelations fall within the confidence bounds (Figure 3.4(bottom)). Figure 3.5 shows the identified local neighborhoods $\mathcal{X}^{(0)}_{local}, \mathcal{X}^{(1)}_{local}$, now in the original data set rather than as residuals $R^{(0)}$ and $R^{(1)}$. As seen in the Figures, one can identify which unit is responsible for modeling a certain part of the data complexity. Two Gaussians units are allocated for the two dominant convex regions in the data set. The procedure has therefore the ability to allow for an interpretation of the allocated units, again see Figure 3.5. As evidenced in Figure 3.4, our test will not admit three units, since no further inference from the ACF can be expected. The final result for the noise-free test data is shown in Figure 3.6 with an RMSE of 0.0075. The complexity of the training data matches the one of the network with two units.

To further demonstrate the robustness of ACF-RAN, we apply the two-unit RBF constructed above to a new data set consisting of the same Hermite polynomial with different noise realization. The resulting ACC function, shown in Figure 3.7, indicates that the model correctly fits the data with only one lag-point outside the confidence interval. Note this test is analogous to cross-validation for ACF-RAN.

The Peaks Data

The "Peaks" data was generated from Equation (3.13) with 1500 training points with $t \in [-50\pi, 50\pi]$. A noisy training set was generated by adding Gaussian noise with 0.5 standard deviations to the z variable. Figure 3.8 shows the RMSE on the test data as training proceeds. As units are allocated the number of autocorrelations that fall outside the limit decreases until the algorithm terminates with 16 units (Figure 3.9). As an additional measure of the decrease of correlated residuals, for each model order, the sum of all autocorrelations over lags h ($\sum_{h=0}^{10} \rho(h)$)is computed and shown in Figure 3.10. In view of Figures 3.9 and

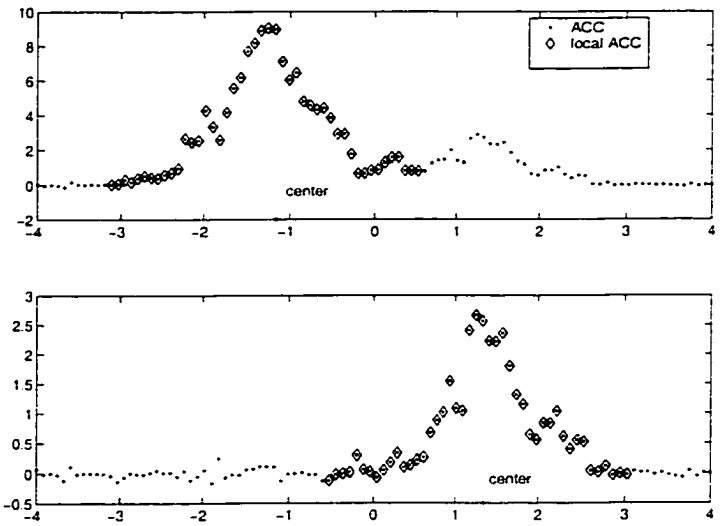


Figure 3.3: ACC during resource allocation of 2 RBF units. The maximal contribution ($\star$) is used as a new center location. The corresponding local training data ($\diamond$) was then used for the adaption of the width and center.

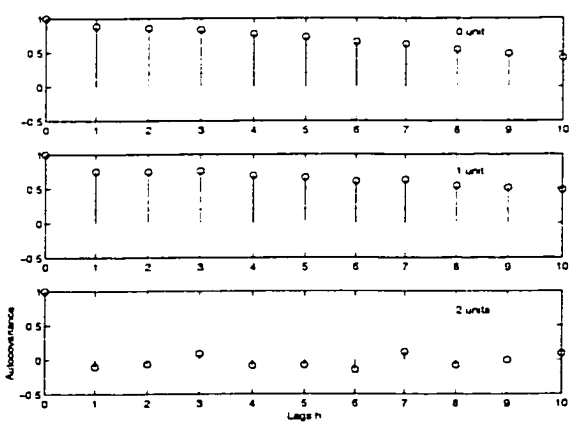


Figure 3.4: As the ACF-RAN algorithm allocates units the ACF decreases until after two units the stopping criteria is achieved and all autocorrelations fall within the confidence interval.

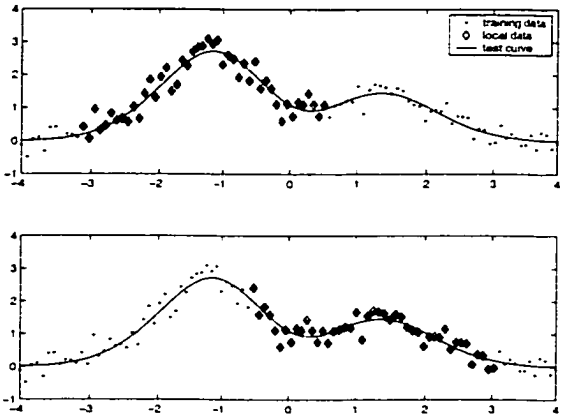


Figure 3.5: The local regions $\mathcal{X}_{local}$ ($\diamondsuit$) (top: $\mathcal{X}^{(0)}_{local}$, bottom: $\mathcal{X}^{(1)}_{local}$) identified during learning with the ACF-RAN on the original data set.

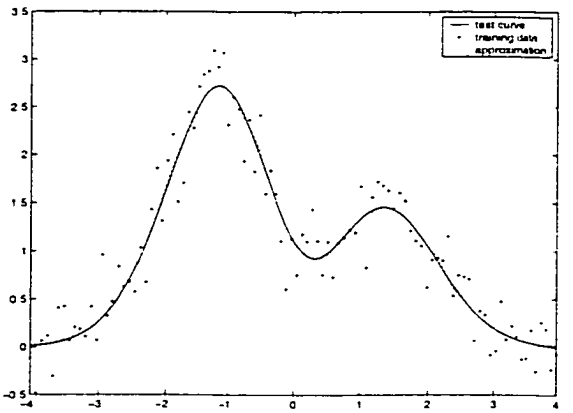


Figure 3.6: Comparison of the ACF-RAN approximation on the noisy data with the noise free test data used in the example.

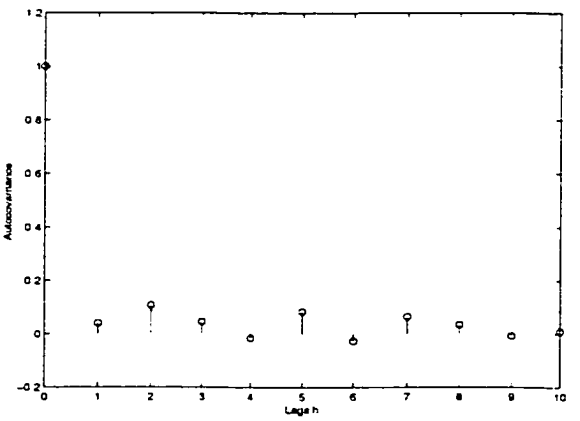


Figure 3.7: ACF function on a noisy test data set. The ACF-RAN reproduces an idd sequence of residuals.

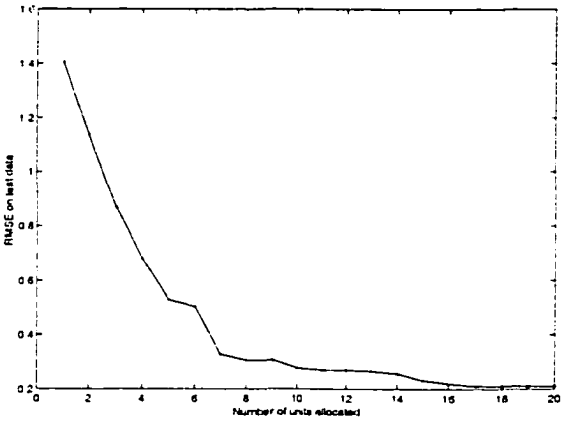


Figure 3.8: The RMSE error on a "Peaks" test set as units are allocated.

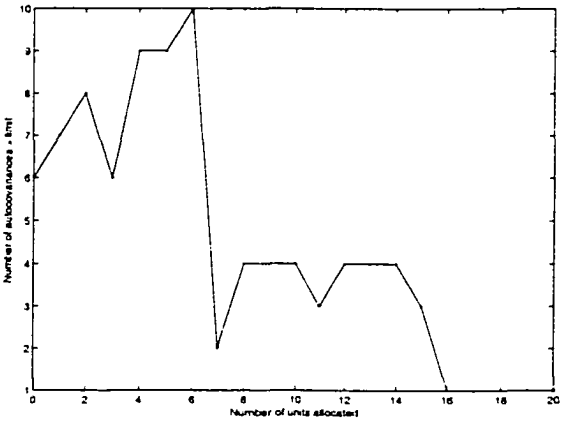


Figure 3.9: Number of autocorrelations that fall outside the confidence limit as units are allocated.

3.10, it appears that six units capture the noisy data well (two autocorrelations outside the confidence limit). As more units are allocated the fine structure is modeled until the network reaches a size of 16 units and no further inference on the residuals can be achieved.

3.5 Summary and Conclusions

In summary, resource allocating networks apply an *ad hoc* stopping criterion for determining the number of required units. Typical RANs employ the model residuals in only a very limited way, i.e., as a means for estimating approximation error. In contrast, our approach focuses directly on evaluating a statistical model

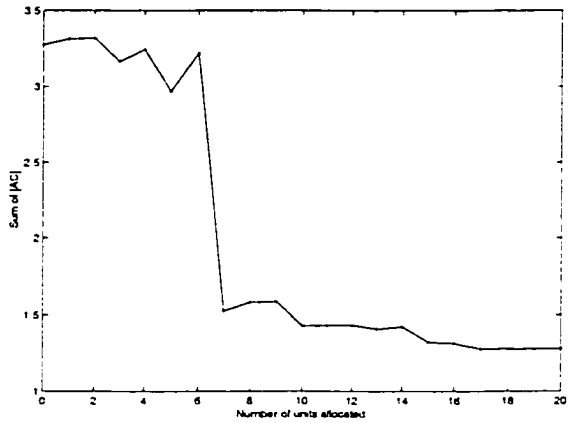


Figure 3.10: A measure of the decrease of the autocorrelations of the residuals for the ACF-RAN.

for the errors. Model validation is a central procedure in any system-identification process in a noisy environment. If the model structure is correct, the residuals should consist of an independent identical distributed (iid) random sequence. To this end, we have introduced a statistical novelty and stopping criterion. The procedure does not rely on the *a priori* assumption that the residuals are drawn from a Gaussian distribution or that an estimate of the covariance matrix for the noise process is available (which is essential in, e.g., [69]). A first and simple model validation test then consists of using the sample autocorrelation of the residuals to check the hypothesis that the residuals are autocorrelated. We use the autocorrelation function to stop adding new units after no inference about the independence of the residuals can be made assuming an iid noise environment. Furthermore we use the contribution to the autocorrelation to identify local regions for network parameter adaptation with a quasi-Newton Method (BFGS). A significant feature of the algorithm is that the resulting RBF network is parsimonious, which results in the possibility of interpretation of the inserted units.

Hence, our algorithm is based on an autocorrelation test applied to the residuals $R^{(K)}$. Generally, in time-series analysis, if the residuals are identified as iid for a model of order K, then one usually applies a higher order autocorrelation test

to check for neglected nonlinearity. In this paper we only make use of the linear residual autocorrelation test. Future work will include higher order correlations and their influence on RANs.

Chapter 4

THE MAXIMUM NOISE FRACTION METHOD FOR FILTERING NOISY TIME-SERIES

We propose a tool for filtering multivariate time series that was initially developed for analyzing multi-spectral satellite imagery. The basic technique, known as the maximum noise fraction (MNF) method [31], may be used to provide a subspace decomposition of a multivariate time series in terms of basis vectors which contain maximum noise (or maximum signal). The methodology is applied to the reduction of data on noisy manifolds [3].

4.1 Introduction

The application of the Karhunen-Loève (KL) procedure (similarly, the singular value decomposition (SVD) or principal component analysis (PCA)) to noisy data can be problematic in that the eigenvectors associated with the largest variance may contain significant amounts of noise. It is well-known that the KL eigenvectors are left unchanged by the addition of white noise while the eigenvalues are all shifted upwards by the variance of the noise [27]. The maximum noise fraction (MNF) method, proposed by [68] (see also [31]), was developed as a noise removal technique for multi-spectral satellite data. Related approaches have been proposed by [1] in the context of singular spectrum analysis as well as for analyzing biomedical time-series via Quotient-SVD [58]. However, our method includes the

estimation of the noise covariance matrix explicitly for multivariate time-series, not considered by other authors.

The MNF technique produces a basis representation, the purpose of which is to separate the noise and the signal as far as possible into distinct subspaces. An interesting feature of this approach is that high-frequency components of the signal are not attenuated as a result of the filtering. Sharp features such as bursts, spikes and non-differentiable points which may provide essential information in the time-series are retained. Additionally, the method lends itself naturally to being applied locally, either in time or space.

4.2 Methodology

Let us consider the observation of a multivariate time signal $x(t) \in \mathbb{R}^q$, sampled P times during $0 \leq t \leq T$

$$x^T(t) = (x_1(t), .., x_q(t)), \quad t = 1 \ldots T,$$

which is composed of a deterministic signal uncorrelated with additive noise at times t ($E[x(t)n(t)] = 0$)

$$x(t) = s(t) + n(t).$$

In terms of data matrices we may write this decomposition as

$$X = S + N,$$

where $X, S, N \in \mathbb{R}^{P \times q}$, e.g., $X^T = [x^{(1)}| \ldots |x^{(P)}], x^{(i)} \in \mathbb{R}^q$, dropping the time index. Unknown are the true signal covariance matrix Σ_s and the noise covariance Σ_n, whereas the covariance of $x(t)$ can be estimated as

$$\Sigma = \langle x(t)x^T(t) \rangle = \frac{1}{T} \sum_{t=1}^{T} x(t)x(t)^T = \frac{1}{T} X^T X.$$

A procedure for estimating the unknown noise covariance matrix Σ_n will be presented in Appendix C (see also Section 4.3). As a result of the uncorrelatedness, we can decompose Σ as

$$\Sigma = \Sigma_s + \Sigma_n.$$

The signal-to-noise ratio of the i-th time series is defined via the respective ratio of signal and noise variances as

$$\mathrm{SNR}_i = \frac{\mathrm{Var}[x_i(t)]}{\mathrm{Var}[n_i(t)]}, \quad i = 1 \ldots q,$$

whereas the noise fraction of the i-th time series is defined as

$$\mathrm{NF}_i = \frac{\mathrm{Var}[n_i(t)]}{\mathrm{Var}[x_i(t)]}, \quad i = 1 \ldots q,$$

the ratio of the noise variance to the total variance for the i-th band. The basic idea, due to [68], is to compute a new basis, ϕ_k, to represent the data with maximum noise fraction

$$\phi_k^{(i)} = \psi_k^T x^{(i)}, k = 1 \ldots q \quad \phi^{(i)} = \Psi^T x^{(i)}, i = 1 \ldots P,$$

resulting in a new representation of X. Writing Φ in its data dependent form, i.e., as a superposition of the data we get

$$\Phi = X\Psi, \quad \Phi = [\phi_1 | \ldots | \phi_k | \ldots | \phi_q], \phi_k \in \mathbb{R}^P \quad \Psi = [\psi_1 | \ldots | \psi_k | \ldots | \psi_q], \psi_k \in \mathbb{R}^q.$$

The basis vectors $\phi_i \in \mathbb{R}^P$ may also be decomposed into signal and noise components, indicted by additional subscripts s, n, as

$$\phi_i = \phi_{i,s} + \phi_{i,n},$$

where $\phi_{i,s} = S\psi_i$ and $\phi_{i,n} = N\psi_i$.

4.2.1 The Optimization Problem

Now the *noise fraction* of the i-th basis vector ϕ_i is defined as

$$\mathrm{NF}(\phi_i) = \frac{\phi_{i,n}^T \phi_{i,n}}{\phi_i^T \phi_i}.$$

This may now be rewritten in terms of ψ_i

$$\mathrm{NF}(\psi_i) = \frac{\psi_i^T N^T N \psi_i}{\psi_i^T X^T X \psi_i} = \frac{\psi_i^T \Sigma_n \psi_i}{\psi_i^T \Sigma \psi_i}. \tag{4.1}$$

We find the vectors ψ_i maximizing the noise fraction by solving the real, *symmetric definite generalized eigenproblem*, which follows from Equation (4.1) after differentiation with respect to ψ_i and setting the result equal to zero:

$$\Sigma_n \psi = \mu \Sigma \psi. \tag{4.2}$$

The resulting generalized eigenvalues $\mu_i, i = 1 \ldots q$ are the noise fractions, corresponding to the generalized eigenvectors ψ_i, ordered such that with increasing μ_i the eigenvectors ϕ_i contain more noise. Thus, given a data matrix X and the solution matrix $\Psi_q = [\psi_1 | \ldots | \psi_q]$ of the generalized singular vector problem (4.2), the orthonormal basis for $\mathbb{R}^q$ is given by $\Phi_q = [\phi_1 | \ldots | \phi_i | \ldots | \phi_q], \phi_i \in \mathbb{R}^P$ where $\Phi_q = X\Psi_q$ and normalization $\Phi_q \Phi_q^T = I$.[1] The set Ψ_q is normalized such that $\Psi_q^T X^T X \Psi_q = I$. Clearly we may express the data without loss as $X = \Phi_q \Phi_q^T X$. Truncating r noise dominated columns of Φ_q acts as a noise filter on X, i.e., the data may be decomposed as

$$X_D = \Phi_D B_D,$$

where the smaller matrix with $D = q - r$

$$B_D = \Phi_D^T X = \Phi_D^T X_D$$

---

[1]The basis vectors are ordered by increasing noise fraction, so a truncation of the basis corresponds to noise filtering.

consists of reconstruction coefficients.

It has been suggested that this procedure may be carried out by applying a whitening transformation based on the covariance matrix of the noise [43] and [1] (see Appendix A). In this setting the resulting problem for determining the basis vectors is a standard SVD rather than a generalized SVD. While this approach is a convenient computational device, we found that the change of basis required to whiten the noise produced an unnatural coordinate system. Hence, all our results are transformed back to the original unwhitened coordinates.

4.3 Estimating the Covariance Matrix of the Noise

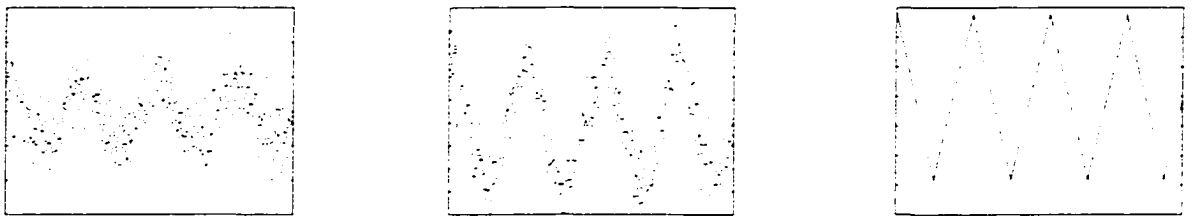
As suggested in [68], the covariance matrix of the noise may be estimated by shifting a time series and computing differences. One can show in this case that the covariance matrix of the differences is approximately twice $N^T N$, see Appendix C for the necessary assumptions. Note that this method requires that the data be smooth, i.e., $x_t \approx x_{t+1}$. It should be noted that the procedure for estimating the covariance matrix of the noise need not be absolutely perfect. The penalty of error is a modest rotation of the subspace spanning the maximum noise fraction eigenvectors.

4.4 Applications

We demonstrate the effectiveness of the maximum noise fraction for filtering noisy data in the context of several illustrative examples.

4.4.1 Filtering Nonsmooth Data

The data in this example was generated specifically to demonstrate the point that nonsmooth functions buried in the data may be accurately recovered. The data set consists of 10 highly cross-correlated noisy time series each consisting of



(a) (b) (c)

Figure 4.1: (a) One of the ten original time series; (b) one term KL reconstruction of (a); (c) one term maximum noise fraction reconstruction of (a).

500 points. The correlation was achieved by mapping three time series to $\mathbb{R}^{10}$ using a random full rank matrix:

1. a sawtooth of amplitude one and period 2π with no noise;

2. a pure noise signal–normally distributed with zero mean and variance 0.01;

3. a squared sinusoid with added noise with zero mean and variance 0.005.

The result of applying the MNF to one of the ten time series is shown in Figure 4.1. The sawtooth is in fact the first MNF basis vector and thus captures the shape of the time series without noise. The KL reconstruction is noisy as the first basis vector contains significant noise. (See, e.g., [39] for a general discussion of the application of KL to data sets.)

4.4.2 The Noisy Circle

A two dimensional circle $x = \cos(\theta)$, $y = \sin(\theta), \theta \in [0, 2\pi]$ embedded in $\mathbb{R}^3$ was used to demonstrate the limitations associated with the KL decomposition in the presence of non-white noise added to the signal (x, y) (see Figure 4.2). We show that with an estimation of the noise covariance (see Appendix C) it is possible to extract the original signals using the MNF transform. The successful discovery of the signal is apparent in view of Figure 4.3. In (a) the KL eigenvectors are shown,

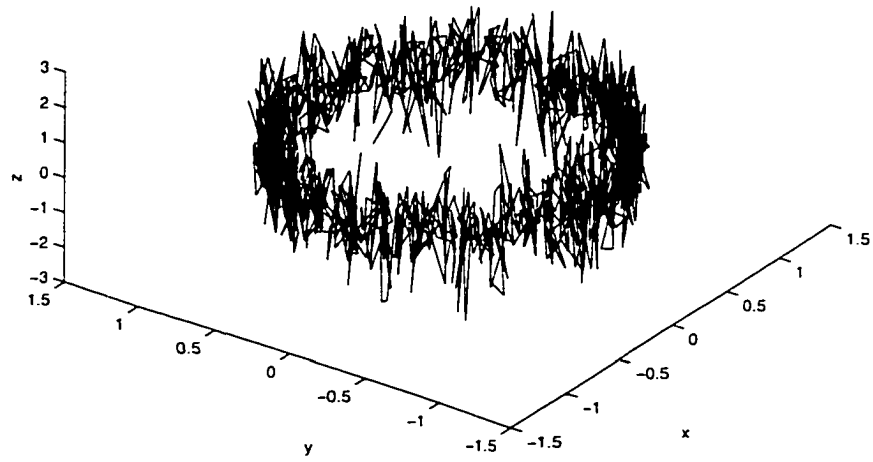


Figure 4.2: A circle in the x-y plane with spatially non-white noise added in three dimensions. The noise variance in z direction is considerably larger than in the x-y plane.

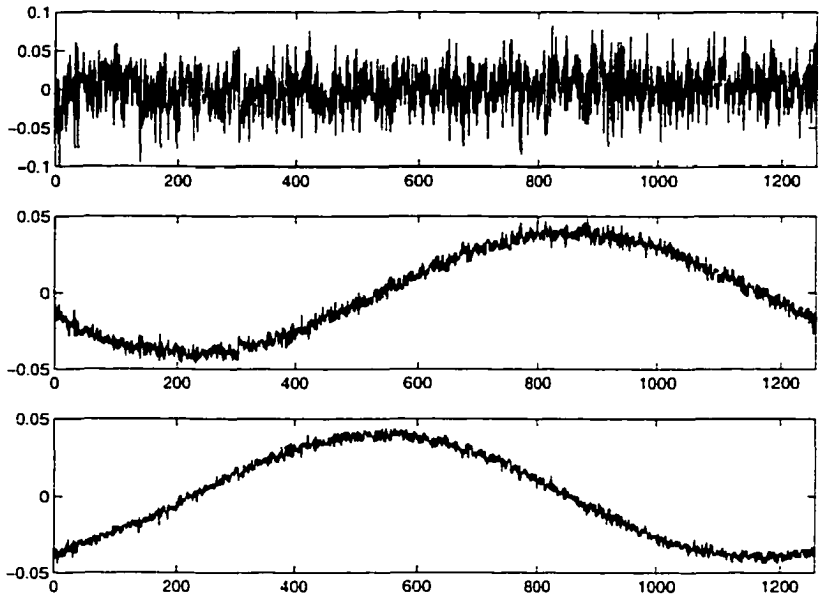
capturing the dominate (noisy) variance directions, whereas the MNF eigenvectors are sorted according to their noise fraction. The signal can be recovered by retaining the first two MNF eigenvectors (in reverse NF order). The signal compared in MNF modes and KL modes is shown in Figure 4.4. Finally, Figure 4.5 shows the full 2-mode reconstructions in $\mathbb{R}^3$.

4.4.3 Multivariate Weather Data

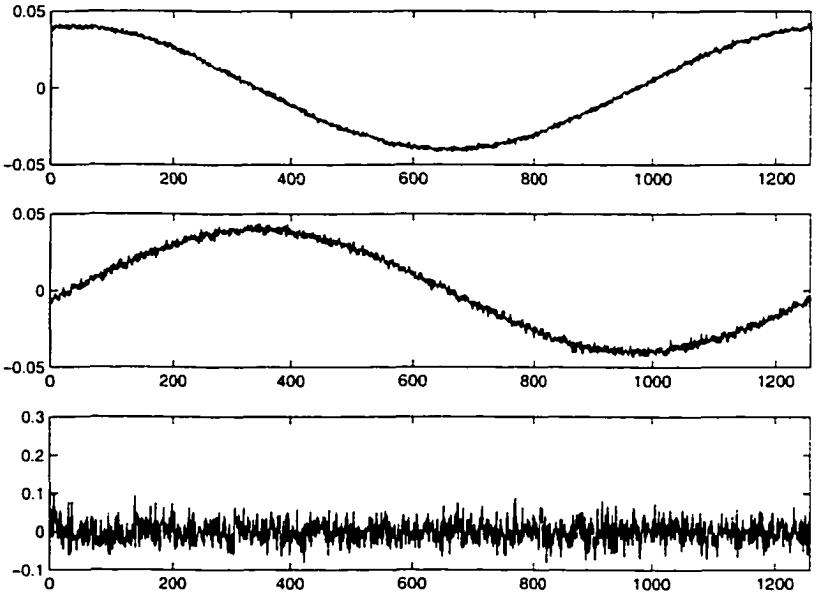
Here we investigate the application of the MNF method to weather data collected during the first four days of October 2000 with a sampling frequency of five minutes. The time series we investigate include temperature, relative humidity, wind speed (average speed), gust speed (speed of the maximum wind) and pressure.[2]

---

[2]This data was made available by the Colorado State University Atmospheric Science Department, Fort Collins, CO. It was collected at the Christman Field (FCC), Foothills Campus Weather Observation Station and may be viewed at www.atmos.colostate.edu/cgi-bin/fcc/form.pl.



(a)



(b)

Figure 4.3: (a) KL eigenvectors: the first mode contains no signal but only noise; (b) MNF eigenvectors based on the estimated noise covariance: the first two modes extract the two-dimensional signal.

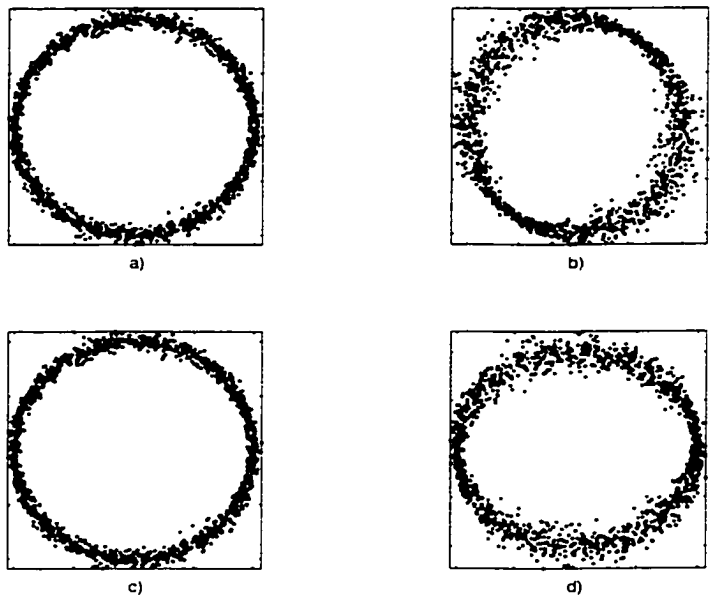


Figure 4.4: (a) Signal in 2-mode MNF space; (b) signal in 2-mode (2nd and 3rd) KL space; (c) 2-mode MNF reconstruction (z-direction); (d) 2-mode KL reconstruction (z-direction).

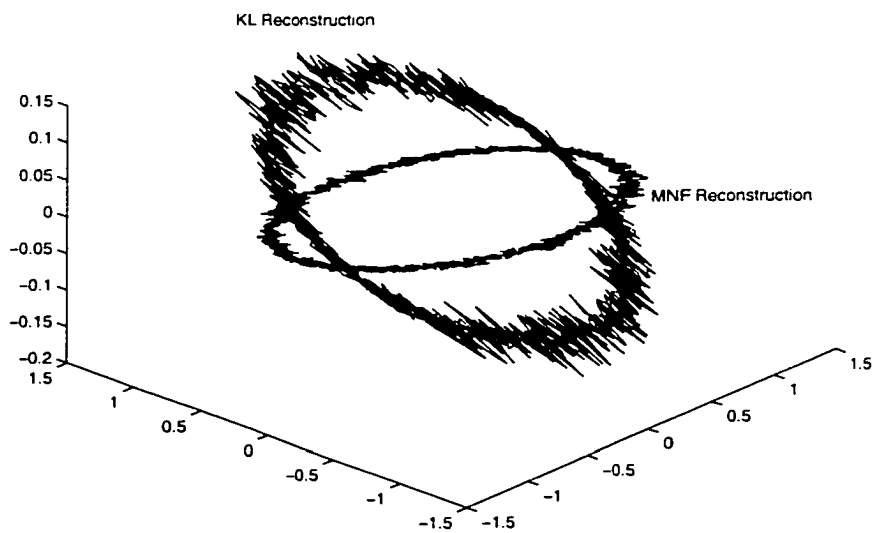
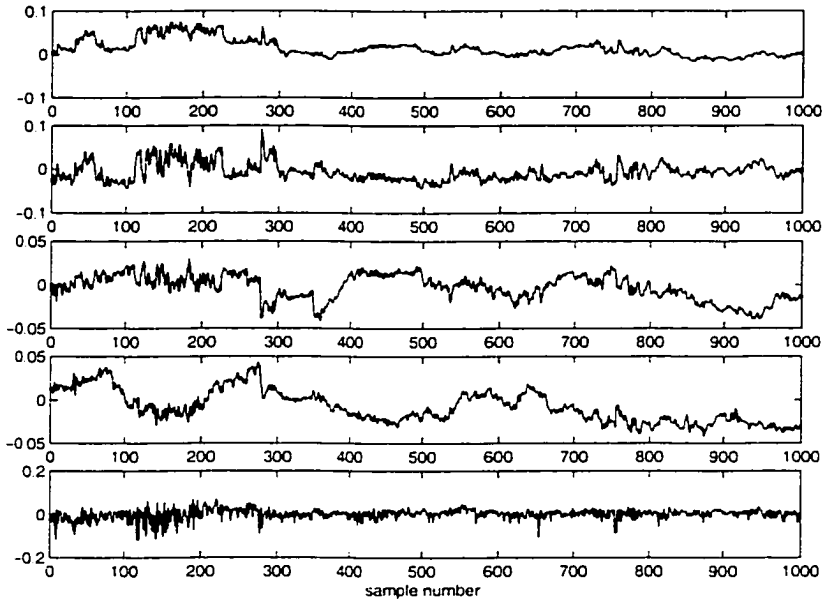
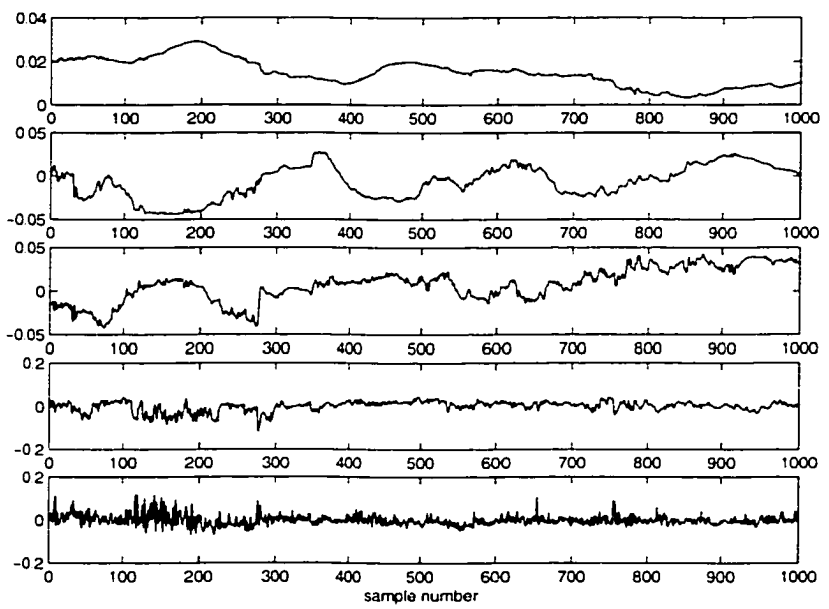


Figure 4.5: 2-mode reconstruction using 2nd and 3rd KL and 1st and 2nd MNF eigenvectors.



(a)



(b)

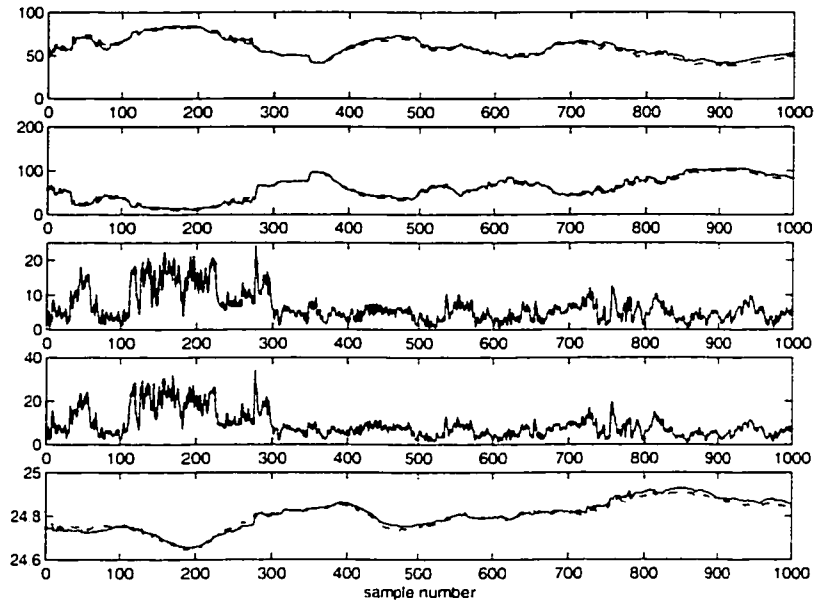
Figure 4.6: Multivariate weather data basis vectors ordered from top to bottom. (a) KL basis; (b) maximum noise fraction basis with maximum signal basis vector at the top and maximum noise basis vector at the bottom.

Again, for purposes of comparison we have included both the KL reconstruction and the MNF reconstruction using the basis vectors shown in Figure 4.6. Using three terms, the KL reconstruction fits each time series accurately; see Figure 4.7. Note that the wind and gust speed time series have high variance but are fit very well. The MNF fit for these time series is smoothed. Note that the temperature, relative humidity and pressure are fit more accurately using MNF than with KL. Thus, the MNF technique fits the low variance data with high accuracy and filters high variance data. As a result, the MNF transformation acts as a smoothing filter, revealing the underlying trends in the wind speed and gust speed time series. This fact is made apparent by the nature of the KL and MNF eigenvectors shown in Figure 4.6. The first few MNF basis vectors containing the signal are much smoother than the corresponding KL basis vectors.

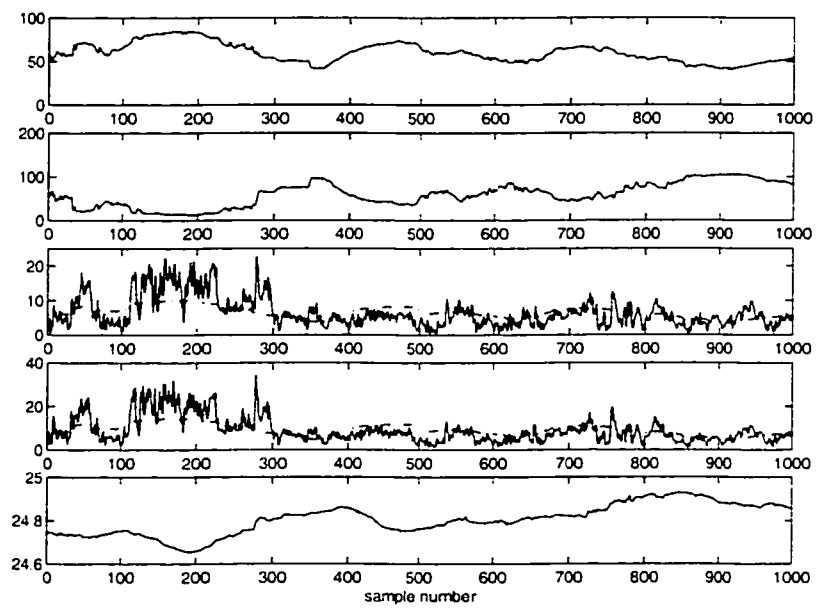
Note that the data was scaled in this example such that each time series has unit variance and zero mean. This is required for the KL procedure, otherwise the dominant eigenvectors span the wind and gust speed subspaces. One advantage of the MNF method is that it is scale invariant.

4.5 Reduction of Noisy Manifolds with MNF

If data is highly correlated in its ambient space it is often possible to construct dimensionality-reducing mappings [39]. In general, if the data matrix has full rank, then linear methods are not able to compress the dimension without loss of data. However, considering the special instance when the data matrix actually contains points that reside, at least approximately, on a manifold, we may employ the WRN [13, 12] to discover any hidden states associated with the data. The application of the WRN architecture to noisy data is challenging and requires special modifications [12]. Here we examine the utility of MNF for use in conjunction with the Whitney Reduction Network (WRN).



(a)



(b)

Figure 4.7: Reconstructions of weather data from October 1-4, 2000 using: (a) three term KL reconstruction; (b) three term MNF reconstruction. The five time-series in each figure measure temperature, relative humidity, wind speed, gust speed and pressure. In each figure the dotted line represents the reconstructed data while the solid line represents the original data.

4.5.1 The Whitney Reduction Network (WRN)

Proposed initially in [13], the WRN provides a tool to express a data set in $\mathbb{R}^q$ globally as the graph of a function. A primary feature of the method is that it provides a good domain for the graph by optimizing on the condition number of the function. Given a data matrix X, it is reduced by projecting onto the d-dimensional domain of the graph. Whitney's Embedding Theorem [33] states that a sufficient condition for this inverse to exist is that the dimension d of this projection satisfy $d \geq 2m + 1$ where m is the dimension of the manifold on which the data lie. The theorem is an idealization that we have found to work well in practice. The parameterization of the data in a given basis $V = [v_1|\ldots|v_d|v_{d+1}|\ldots|v_q] = [V_1|V_2]$ in terms of the reduced coordinates $\hat{p} = V_1 x$ where the representation of a point x, written $\tilde{x}$, is given by

$$\tilde{x} = V_1\hat{p} + V_2 f(\hat{p}), \tag{4.3}$$

where V_1 is a basis for the domain and V_2 is a basis for the range. In this setting the data may now be viewed as the graph of a function $(\hat{p}, f(\hat{p}))$. Alternatively, the first term $V_1\hat{p}$ may be viewed as the *linear reconstruction* while the second term $V_2 f(\hat{p})$ is the *nonlinear reconstruction*; see [39] for further details and examples of this procedure.

4.5.2 The WRN with the Maximum Noise Fraction Transformation

It has been observed [12], perhaps not surprisingly, that the presence of noise in data may impede the search for a good projection. Thus it is natural to investigate the utility of the maximum (equivalently, minimum) noise fraction method. We shall see that this approach has considerable appeal given it naturally separates the signal and noise into orthogonal subspaces. Recall that the MNF noise filtered data may be written

$$X_D = \Phi_D B_D.$$

We propose to compress X_D by parameterizing the last $D-d$ columns of Φ_D by the first d columns. This may be achieved by decomposing Φ_D using (4.3)

$$\Pi = V^T \Phi_D.$$

Following [13], as a preprocessing step the basis of the row space of Φ_D is changed to improve the condition number of the reconstruction. In practice, it is the first d columns of Π that serve as the domain of the graph, while the $D-d$ columns serve as the target of the domain, i.e., the range, following (4.3). A radial basis function expansion is employed to approximate this map [13].

The complications associated with the construction of a basis V for a well-conditioned projection are addressed in Appendix C. See Figures 4.8 and 4.9 for a summary of the decomposition and reconstruction procedures. The data modeling stages using the maximum noise fraction transform are as follows.

- Perform a MNF transform, determining Φ_D and B_D for the compression of X using only signal dominated vectors Φ_D. Note that the signal-to-noise ratios for each ϕ_i are given in terms of eigenvalues μ_i

  $$\mathrm{SNR}(\phi_i) = \frac{1}{\mu_i} - 1,$$

  and indicate how many terms in Φ are to be retained.

- Find a basis $V = [V_1|V_2]$ to construct Π, optimizing the conditioning of the inverse map. The details associated with the construction of a well-conditioned inverse will be addressed in Chapter 5. The projector Π_d is used to construct the domain and $\Pi_d^{\perp}$ for the target.

- Approximate the map from the parameterized domain to the target employing a nonlinear function estimation, e.g., a neural network; this leads to the approximation of Π by $\tilde{\Pi}$ (see Figure 4.9).

- Use V to return to the MNF representation and B_D to reconstruct the approximation of the data:

$$\tilde{X}_D = \tilde{\Phi}_D B_D = \tilde{\Phi}_D \Phi_D^T X.$$

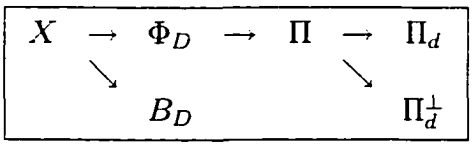


Figure 4.8: A summary of the decomposition and parameterization of a data set using the WRN with MNF.

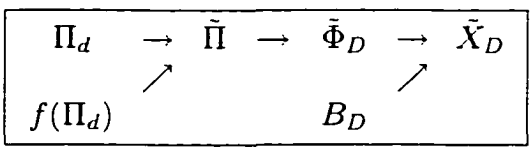


Figure 4.9: A summary of the reconstruction of a data set using the WRN with MNF based on the decomposition shown in Figure 4.8. The tildes denote that the quantities are now the approximations (to the true values) as produced by the RBF fitting procedure.

4.5.3 A Noisy Space-Time Signal

The signal investigated here is a pulse moving with unit velocity on a circle: $x(h-t)$, h representing the spatial variable and t time; where $x(\theta) = x(\theta + 2\pi)$ [12]. Without noise this data is topologically a circle (a one-dimensional manifold) in a high dimensional space. The pulse x is simulated as a traveling Gaussian (see Figure 4.10). If a noise process $n(t,h)$ is added the space-time symmetry has been broken (see Figure 4.12)

$$x(h,t) = \exp -(t-h^2)/\gamma + n(h,t).$$

With $\gamma = 25$, the function was sampled at 64 points in the h-direction at half-integers and at 256 points in t-direction. The resulting data matrix has the format

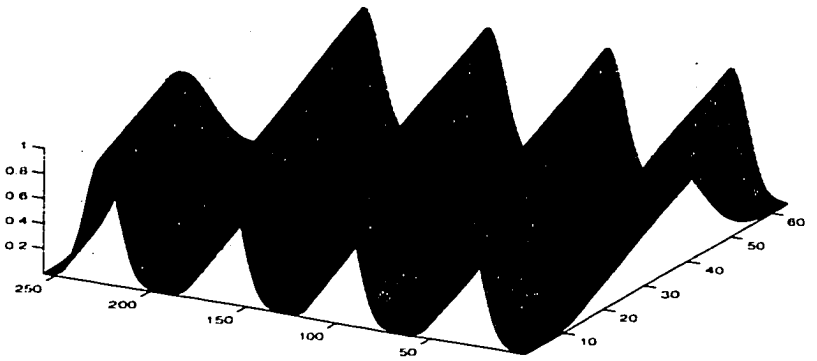


Figure 4.10: Noise-free traveling wave. Sampled at 64 points in spatial direction h and 256 points in time t.

$X \in \mathbb{R}^{256\times 64}$. The data sits now only approximately on a circle. Another way to look at the data is to take 64 spatial bands each one 256 long in time. The noise that was added to the signal was normally distributed with variance $\Sigma_N \in \mathbb{R}^{64\times 64}$ (see Figure 4.11); the noise is assumed to be temporal white such that $n \sim N(0, \Sigma_N)$. The data decomposition using MNF and the WRN as illustrated in Figure 4.8 is now employed to filter and represent the data. The separate reconstructing of the linear and the nonlinear component to the original data space are shown in Figure 4.13 and 4.14, respectively. The MNF reconstruction without employing the WRN, $X_D = \Phi_D B_D$ with D=6, retaining 6 MNF modes out of 32 bands is illustrated in Figure 4.15. Employing the WRN, the parameterization of Φ_D using a 2 dimensional basis $V_1 = [v_1|v_2], v_i \in \mathbb{R}^6$ results in a 5% lower reconstruction error, compared with the MNF based reconstruction to the original space-time signal, as shown in Figure 4.16.

4.5.4 Predicting a Change in the Weather

Here we consider an application of the WRN with MNF filtering to a set of six time-series measuring weather variables, described in the previous section for the month of October, 2000 as shown in Figure 4.17. (Note that the noise fraction basis was calculated over the same interval as the training data.)

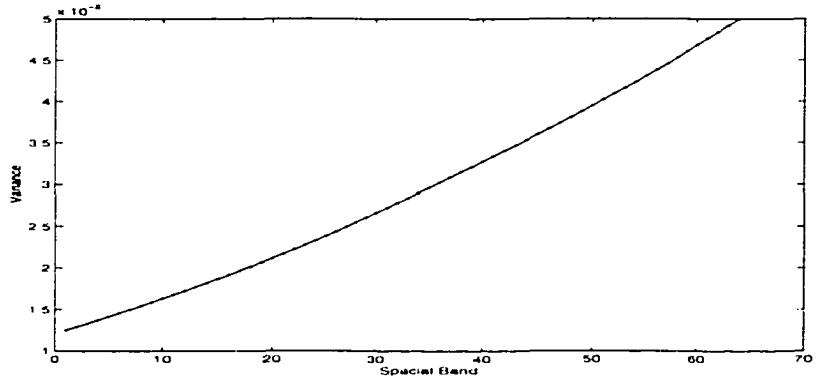


Figure 4.11: Noise variance as a function of each of the 64 spatial bands.

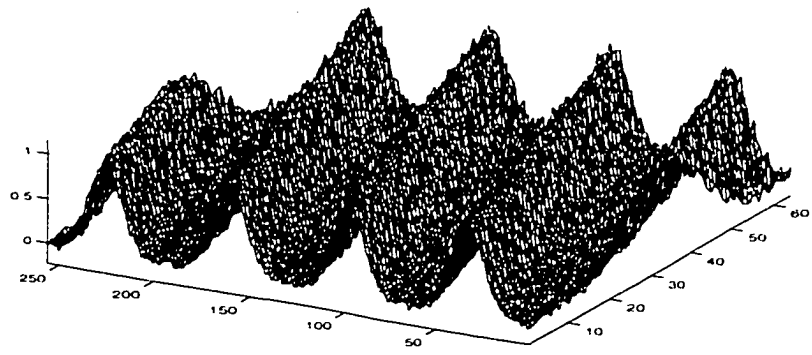


Figure 4.12: The Traveling wave corrupted with spatially non-white noise.

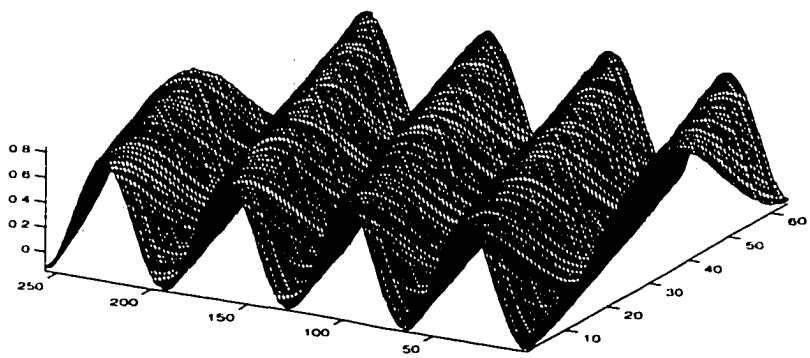


Figure 4.13: Linear reconstruction of 6-mode (D=6) MNF projection based on 2-dimensional secant basis (d=2).

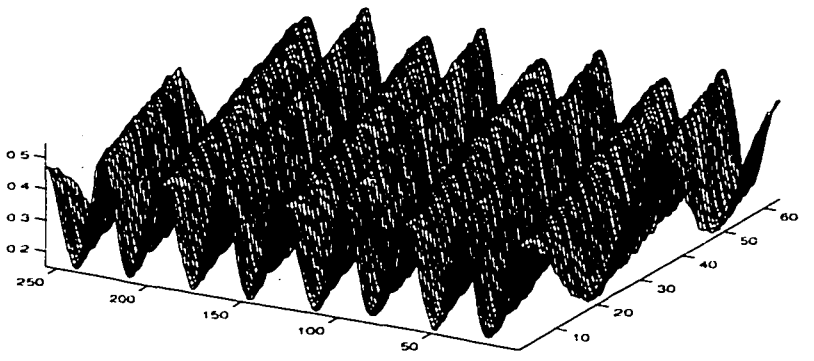


Figure 4.14: Nonlinear reconstruction of 6-mode MNF projection based on 2-dimensional secant basis.

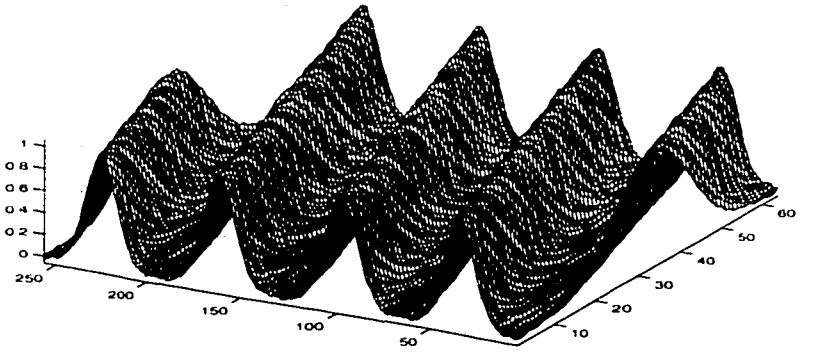


Figure 4.15: Reconstruction of 6-mode MNF filtered traveling wave, without the WRN modeling step.

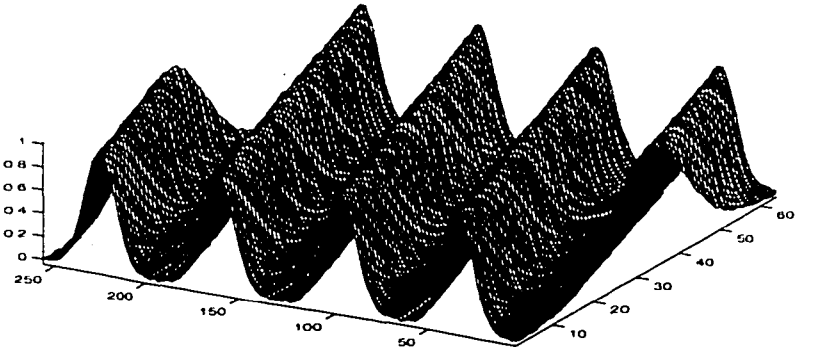


Figure 4.16: Full reconstruction of 6-mode MNF filtered traveling wave with nonlinear mapping from 2-dimensional secant basis to 4-dimensional orthogonal residual.

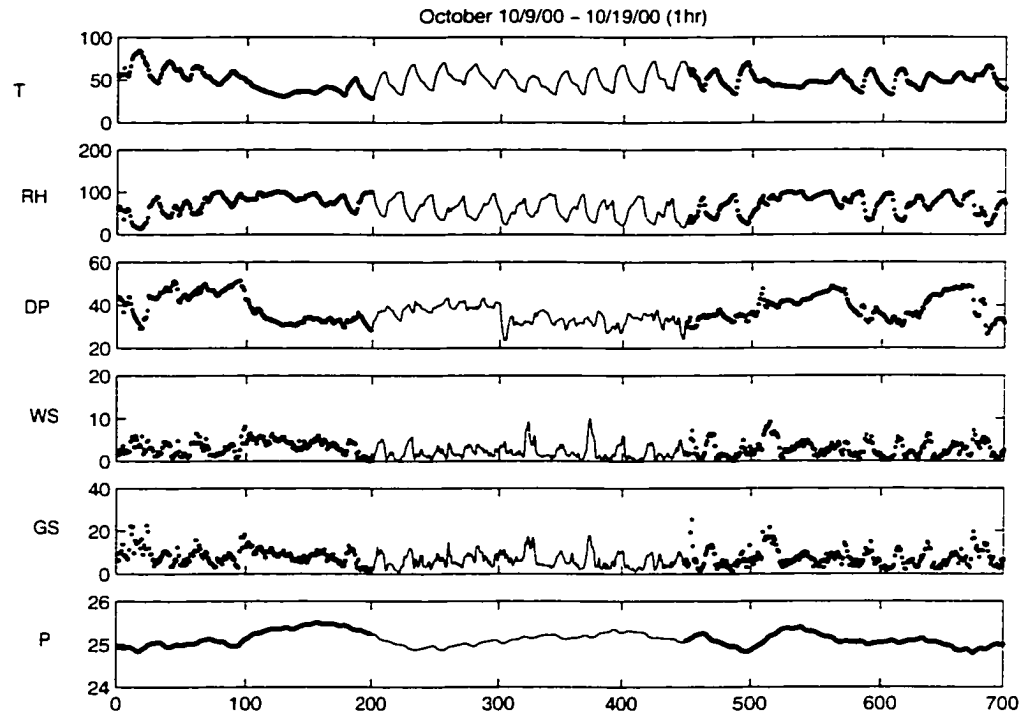


Figure 4.17: Raw weather data consisting of temperature (T), relative humidity (RH), dew point (DP), wind speed (WS), gust speed (GS) and pressure (P) collected over the month of October, 2000 at hourly intervals. The dark points represent testing data while the light points from October 9–19 were used for building the radial basis function model.

We found that choosing a filter corresponding to $D = 4$ eliminated what appeared to be correlated noise. Further, we employed $d = 3$ so the radial basis function was required to empirically model a function

$$f : \Pi_3 \subset \mathbb{R}^3 \rightarrow \Pi_3^{\perp} \subset \mathbb{R}.$$

We selected a period of ten days, October 9–October 19, for constructing the model using the method of orthogonal least squares (OLS)[16].

At approximately point 475 we detect that the model residual has become significant (see Figure 4.18) indicating that the underlying weather pattern is changing. Note that this point in time occurred 25 hours after the end of the training data. Furthermore, at about point 510, or a day and a half after the

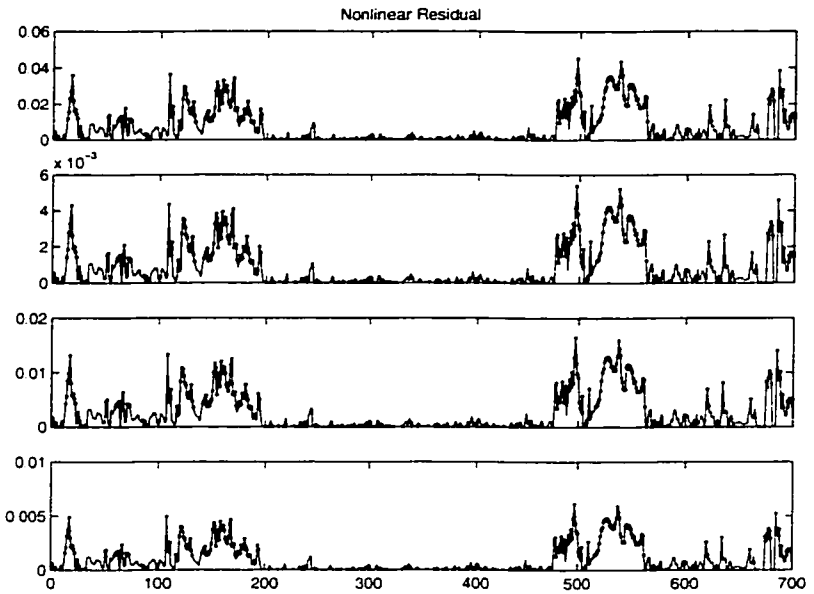


Figure 4.18: Jump in nonlinear residual indicates an impending change in the weather. Points marked with a dot indicate that the magnitude of the residual exceeded the maximum residual of the training set.

model detected a change, there was an actual shift in the weather pattern as indicated by the nonoscillatory temperature profiles. Thus, this example suggests that the model predicted a change in the weather roughly a day before it happened. Figure 4.19 visualizes the complete changes in the weather pattern, based on the nonlinear residuals for the fully reconstructed weather time series.

This study is meant only to be an illustrative example of the capacity of this reduction architecture when employed with the MNF methodology.

4.6 Relationship to Independent Component Analysis

The task of filtering noise from data may also be viewed as a problem in multiple source separation. In particular, correlated noise may have significant structure, to the point that it may be interpreted as another signal. Thus, it is interesting to consider how algorithms for source separation such as independent component analysis (ICA) [17] perform on the weather data.

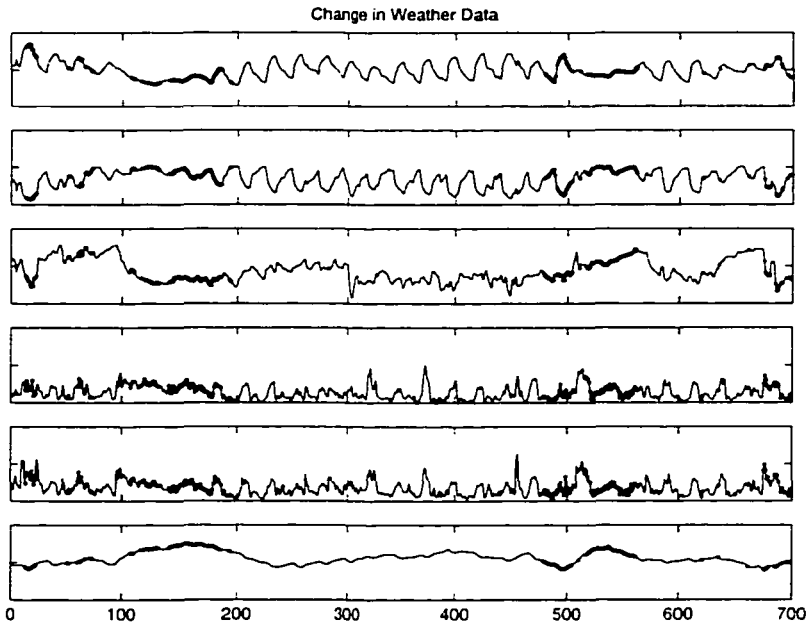


Figure 4.19: The predicted change in the weather data, based on the nonlinear residuals (see Figure 4.18). Dark points indicate a change in the weather pattern.

In independent component analysis it is assumed that the observed signal $x(t)$ is the result of a linear transformation $A \in \mathbb{R}^{q \times q}$ of independent source signals

$$x(t) = As(t),$$

where the independence condition may be formulated in terms of marginal probabilities

$$p(s) = \prod_{i=1}^{q} p_i(s_i).$$

i.e., the multivariate probability distribution of $s(t)$ is factorizable. Now, given $x(t)$, the task is to recover the independent sources $s(t)$; this may be achieved using higher order moments or cumulants in the general framework of information theory and neural networks [20]. To compare the MNF algorithm with ICA we applied both techniques to the multivariate weather data described in Section 4.2. The results are shown in Figure 4.20. Perhaps surprisingly, there is considerable similarity between the resulting MNF eigenvectors and independent components.

The first MNF eigenvector corresponds closely to the pressure variable as does one of the basis vectors resulting from the FastICA routine [36]. The second MNF eigenvector is similar to independent component number two and neither correspond to a physical variable. The third MNF eigenvector provides the least correspondence to the independent components, but by elimination corresponds most to independent component number five; both manifest oscillations of similar period. The fourth MNF eigenvector corresponds closely to independent component number four; again, this is not a physical variable. The fifth MNF eigenvector corresponds closely to the wind/gust speed as well as to independent component number one. The last MNF eigenvector is the same as independent component number three and is non-physical.

Hence, we conclude that there are considerable similarities between the two methods, indeed, the independent components appear to be a small rotation of the MNF basis. Despite these similarities, there are some fundamental differences worth noting. Firstly, the MNF method orders the resulting basis vectors. Thus, it was clear in our application in Section 4 which vectors should be truncated; this is not the case with the ICA representation. Additionally, the ICA algorithm is much more involved than the solution of the generalized singular vector problem. Actually, there are many different algorithms for computing independent components. To check our work we applied the Jade algorithm [14] for ICA and the results were very similar, but not identical. Lastly, it should be noted that in general ICA cannot be used to distinguish signals from multivariate Gaussian noise, but rather to analyze independent sources and their contribution to the original signal [17].

4.7 Conclusions

The MNF technique provides a useful subspace decomposition for noisy as well as high-variance time series. Unlike other applications of this type of method

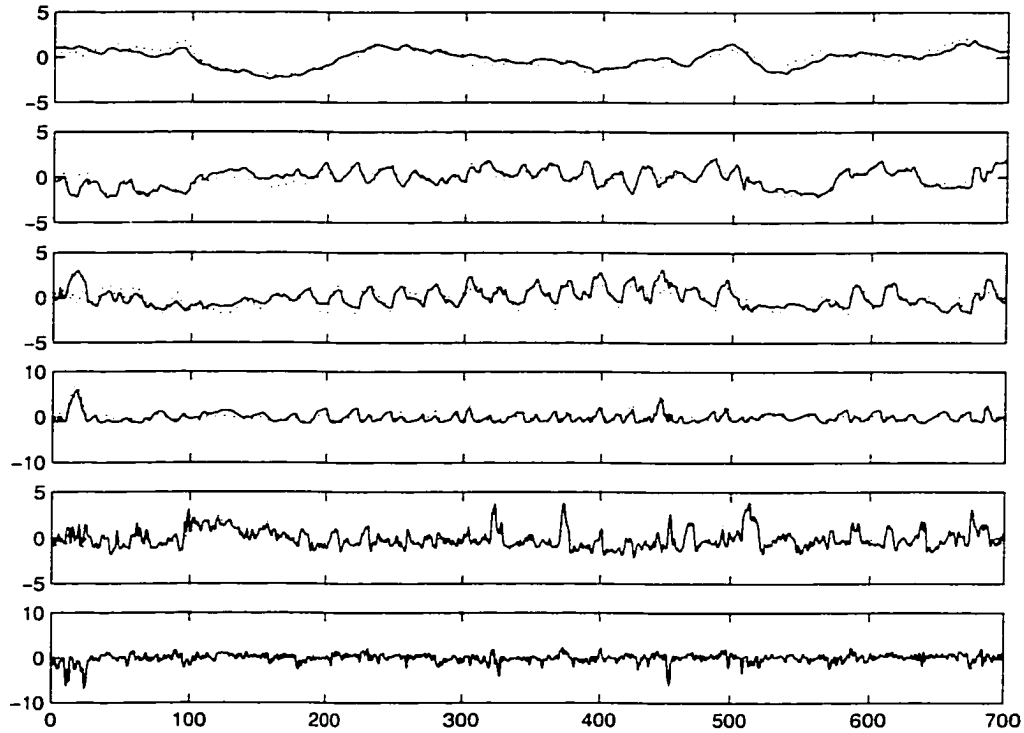


Figure 4.20: A comparison of the results of applying the MNF method and ICA to the weather data. The solid lines correspond to the maximum noise fraction eigenvectors ϕ_i, ordered from top to bottom with increasing noise. The independent components most similar to the MNF eigenvectors are plotted with dotted lines.

to time series, here the covariance matrix of the noise was estimated, following [68], by computing differences of shifted vectors. The method was shown to be effective for filtering both nonsmooth data and data with high-variance bands. In addition, the MNF was integrated with the Whitney Reduction Network [13] –a tool for data reduction–as an effective means to reduce noise. Lastly, a comparison of the MNF method to ICA was presented. The results indicate, at least for the climate data considered here, that there are some surprising similarities between MNF and ICA. This will be the subject of further investigations.

Chapter 5

SECANTS AND GOOD PROJECTIONS

In Chapter 4 we applied the Whitney Reduction Network (WRN) to a high-dimensional data set. The WRN constructs a parameterization of the data, implemented as an invertible projection onto a lower dimensional manifold. Finding such projections is a challenging task. In this chapter we discuss several methods that can be employed to find good quality projections, which are characterized by optimizing the trade-off between providing a low dimensional description of the data and allowing the construction of a well-conditioned inverse. A good quality projection will then be applied to a medical data set.

5.1 Data Parameterization Via Projections

Assume that a data point $x \in \mathcal{A} \subset \mathbb{R}^q$ of a data set $\mathcal{A} = \{x^{(\mu)}\}, \mu = 1 \ldots P$ was obtained by sampling a m-dimensional submanifold $\mathcal{M}$ of $\mathbb{R}^q$. Note that q, the ambient dimension, may be quite large, e.g., the number of pixels in an image or the number of sensors in a physical experiment, whereas it is possible that m, the intrinsic dimension, is much smaller. The motivation for finding good projections is based on the reduction mapping which is the essential first step in the WRN [12]. If a point $x \in \mathcal{A}$ is decomposed under a projector $\mathbb{P}$

$$x = \mathbb{P}x + (I - \mathbb{P})x, \tag{5.1}$$

with $p = \mathbb{P}x$ and $q = (I - \mathbb{P})x = \mathbb{Q}x$, then Equation (5.1) gives a decomposition of x in $p \in \mathcal{R}(\mathbb{P})$, lying in the range of $\mathbb{P}$ and $q \in \mathcal{N}(\mathbb{P})$, in the null space of $\mathbb{P}$. Following Whitney's Embedding Theorem [33], if the rank($\mathbb{P}$) $= d > 2m$, then the existence of a global map from the range of the projector to its null space $q = f(p)$ is guaranteed and we may then parameterize the data set $\mathcal{A}$ in terms of p via

$$x = p + f(p). \tag{5.2}$$

Using an orthogonal basis in $\mathbb{R}^q$, $V = [v_1|v_2|\ldots|v_q]$, a rank-d orthogonal projector $\mathbb{P}$ can be constructed by $\mathbb{P}_d = V_d V_d^T$, where $V_d = [v_1|v_2|\ldots|v_d]$. The coefficients of $p = \mathbb{P}x$ are given by $V_d^T x$ and provide the d-dimensional representation. Since m is generally unknown, an appropriate basis V_d for $\mathbb{P}$ must be estimated empirically from the data set $\mathcal{A}$; this is the topic of this chapter.

5.2 Good Projections

Given the data set $\mathcal{A}$ we need to find a good projection in order to parameterize x using Equation (5.2). An acceptable projection will not "collapse" any pair of data points in the range of $\mathbb{P}$, that is, only projections are admissible that do not project along a secant connecting any two pairs of points in $\mathcal{A}$. Furthermore, the reconstruction of the orthogonal component $\mathbb{Q}x$ will employ a nonlinear function approximation for the target $q = f(p)$. This inverse mapping should be as well-conditioned as possible to achieve good generalization. For example, the stability of the linear weights w_i in a RBF network are directly related to the conditioning of the inverse map; an ill-conditioned mapping will result in the amplification of possible noise contained in the measurements and result in a larger variance of the linear weights w_i. This effect is known as variance inflation [7] and can be dramatically reduced if the condition number of the mapping to be approximated is controlled. A *good quality projection* will provide us therefore which a parameterization of the data such that d is as low as possible, whereas the inverse is as

well-conditioned as possible. The action of the projector $\mathbb{P}$ on two distinct points $x, y \in \mathcal{A}$ is therefore restricted to satisfy the following inequality, as proposed in [13, 12]

$$||\mathbb{P}x - \mathbb{P}y|| \geq k^*||x - y|| \quad x, y \in \mathcal{A}, \tag{5.3}$$

where $k^* > 0$ (typically $k^* \in [0.1, 0.5]$) is a fixed user-supplied tolerance, measuring the maximal allowed distance of any two data points in the range of $\mathbb{P}$. We now seek a dimension d of the range of $\mathbb{P}$ as small as possible satisfying Inequality (5.3) for a given k^*. Using the set $\Sigma = \{\hat{k}_1, \ldots, \hat{k}_N\}$ of unit secants

$$\hat{k} = \frac{x - y}{||x - y||} = \frac{k}{||k||}, \quad x \neq y, \tag{5.4}$$

stored in the matrix $K \in \mathbb{R}^{N \times q}$, where $N = P(P-1)/2$, counting each data pair only once, then Equation (5.3) may now be formulated as

$$||\mathbb{P}\hat{k}|| \geq k^* \quad \forall \hat{k} \in \Sigma, \tag{5.5}$$

with the definition of the minimum projected secant norm

$$k_{\min} = \min_{\hat{k} \in \Sigma} ||\mathbb{P}\hat{k}||,$$

condition (5.5) may be expressed as

$$k_{\min} \geq k^*.$$

Note that $k = 1/k_{\min}$ is the Lipschitz constant of the inverse map $\mathbb{P}^{-1}$ and should be as small as possible for a well-conditioned map. The minimum secant norm $k_{\min}$ and the distribution of k may be utilized to asses the quality of projections and to improve the basis V_d until an acceptable value of $k_{\min}$ has been found.

As an illustrative example we consider the pringle curve[1], defined as

$$\theta \rightarrow (\sin\theta, \cos\theta, \sin 2\theta).$$

---

[1]Note that Whitney's embedding theorem does not apply here. A projection onto $\mathbb{R}^2$ will generally not preserve the differential structure of the manifold $\mathcal{M}$. This example was initially considered in [13].

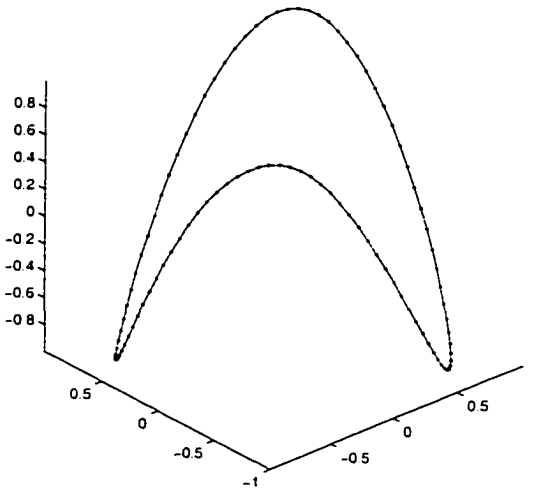


Figure 5.1: The pringle curve is a one-dimensional manifold embedded in $\mathbb{R}^3$.

and sampled on 100 data points with $\theta \in [-\pi, \pi]$ (Figure 5.1). Figure 5.2 shows the set of unit secants of the pringle if the ordering of points is maintained along the curve. This introduces a bias in the unit secant distribution due to the single counting of pairs, as can be observed if we compare the set Σ. computed from the same but randomized data set (Figure 5.3).

5.2.1 The Secant-SVD Basis

A basis V_d for the construction of a good projector $\mathbb{P}$ can be found by considering the principal components, identified with the left singular values, of the unit secant matrix K, obtained by performing a singular value decomposition (SVD) [30] of K

$$K = USV^T,$$

where U is $N \times q$ matrix with orthogonal columns, V is a $q \times q$ orthogonal matrix, and S is a $q \times q$ diagonal matrix which has the singular values of K as its entries. Following the argument in [13, 12] , if $\text{rank}(K) = r < q$, then the projection along any vector $\hat{v}$ in the $\text{span}\{v_{r+1}, ..v_n\} = \text{null}(K)$ associated with zero singular values will result in $\mathbb{P}_{\hat{v}}\hat{k} = \hat{k}$ and consequently $k_{\text{min}} = 1$ allowing us to chose

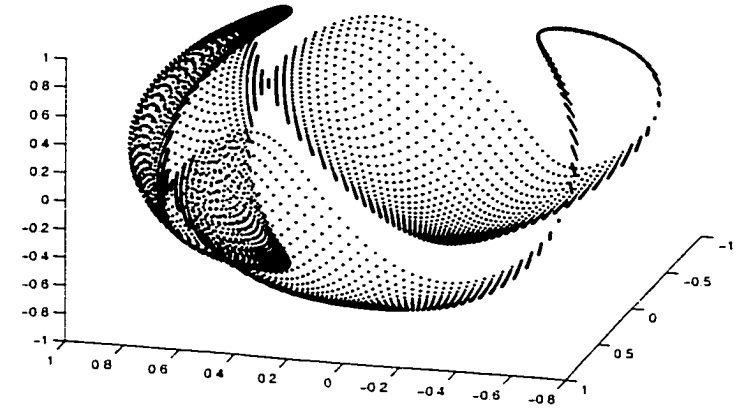


Figure 5.2: The unit secants of the oriented pringle curve where $\mathcal{A} = \{x(\theta_1), \ldots, x(\theta_{100}) | \theta_i < \theta_j, i < j\}$.

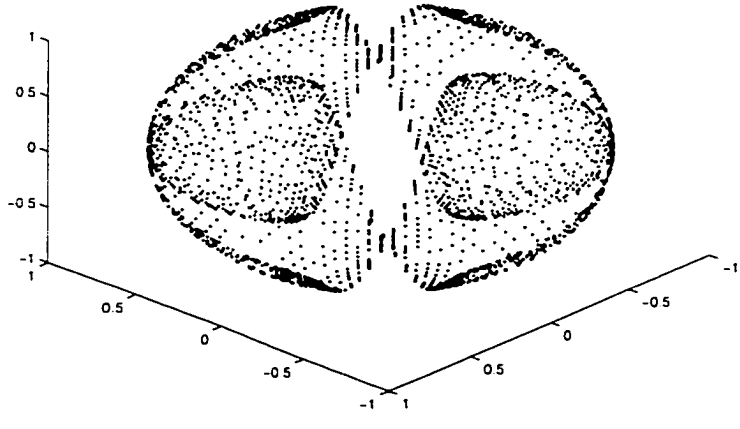


Figure 5.3: The unit secants of the pringle curve computed from the randomized data set $\mathcal{A}$. The randomization results in additional directions of unit secants, which were neglected if an ordering of the points on the curve is maintained.

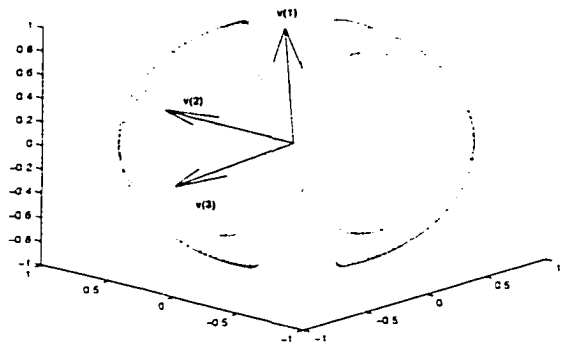


Figure 5.4: SVD directions of the set of unit secants for the pringle data. The principle directions $v(1)$, $v(2)$ and $v(3)$ are associated with the left singular values in decreasing order.

$k^* = 1$. Expanding this argument it is expected that a projection along directions associated with small singular values will result in a larger minimum secant norm $k_{\min}$ and a selection of a larger k^*, on average at least. Applying the previous argument, the columns of K are projected onto the d-dimensional eigenspaces spanned by the principal components of K, $V_d = [v_1|v_2|\ldots|v_d]$. The smallest dimension d for which Equation (5.5) is satisfied determines the good projector $\mathbb{P}_d$.

For the pringle example, Figure 5.4 shows the resulting principal directions of the unit secant matrix K. As can be observed the principal direction associated with the largest singular value, $v(1)$, points in the north-south direction, which is the direction along we would expect to project to obtain a well-conditioned inverse (see Figure 5.5 for such an (invertible) projection). This observation is contrary to the argument made above and a result of the non-uniformity of the unit secant distribution, which is considerable larger towards the poles as evidenced in the histogram of the norms of the projections of K onto $v(1)$, Figure 5.6. It can be shown analytically that secants reveal the north (south) pole at the direction that should be projected along [13].

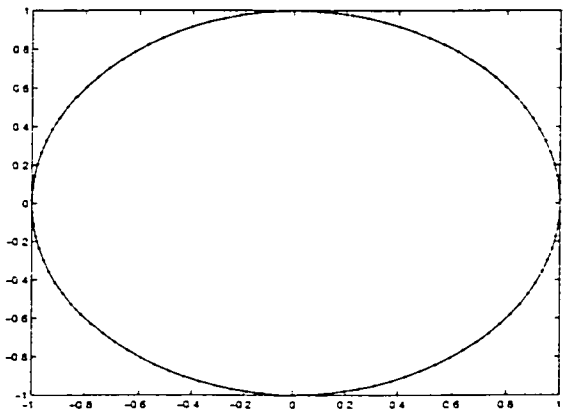


Figure 5.5: The pringle data projected onto the x-y plane spanned by $v(2)$ and $v(3)$.

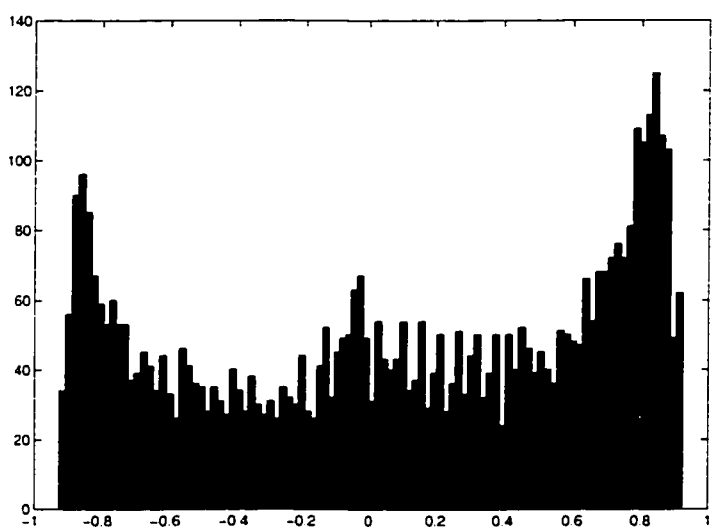


Figure 5.6: Histogram of the norms of the projections of K along the first principal direction, v(1), of unit secants.

5.2.2 The Adaptive Secant Algorithm

Following the previous example, Figure 5.8 shows the principal component associated with the smallest singular value of K, $v(3)$, for the construction of $\mathbb{P}_2 = (I - v(3)v^T(3))$, a projection onto the 2 dimensional subspace spanned by $v(1), v(2)$ (Figure 5.4). This direction is identified as a bad direction due to the large number of secants that do not satisfy Equation (5.5) (denoted by circles in Figure 5.8). This set of secants is referred to as *bad secants* and collected in a set $S = \{\hat{k} \in \Sigma \mid ||P_{d-1}\hat{k}|| < k^*\}$. As proposed by [13], the set S may be used in an iterative procedure to update the initial unit secant covariance matrix $\Theta = K^T K$ $(q \times q)$, with the goal of rotating the eigendirections in a manner to reduce the number of bad secants. A trace preserving update of Θ, weighting S by a constant α, is employed to achieve this

$$\Theta' = (1 - \frac{\alpha}{N})\Theta + \frac{\alpha}{m} S^T S, \tag{5.6}$$

where m is the number of bad secants contained in S. Geometrically, this weighting will iteratively pull the principal directions associated with large singular values into the direction of bad secants and those eigendirections that correspond to small singular values into directions along which we expect good projections, i.e., regions on the d-dimensional unit sphere, where few or no secants are found. This procedure is specified in Algorithm 5.1.

The adaptive secant algorithm as presented in [13, 12] seeks a single projection that eliminates components that lie in a r-dimensional subspace that has no direction colinear with a secant of $\mathcal{M}$, $\mathbb{P} : \mathbb{R} \rightarrow \mathbb{R}^{q-r}$. The projection is constructed by

$$\mathbb{P}x = (I - v_1 v_1^T - v_2 v_2^T - \ldots - v_r v_r^T)x. \tag{5.7}$$

This procedure can be problematic and prevent the algorithm from extracting a small dimension $d = q - r$, necessary to describe the intrinsic nature of the data.

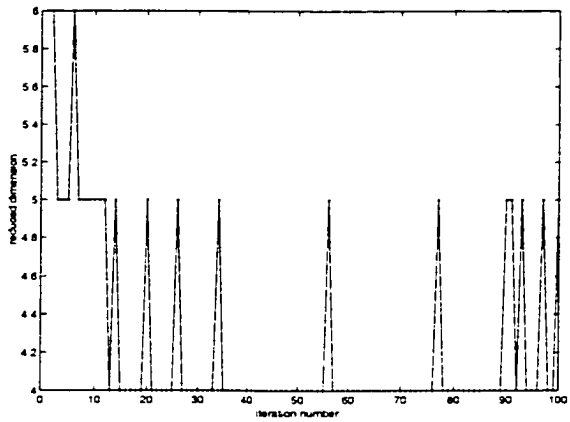


Figure 5.7: The admissible projection dimension as a function of the iteration number for the adaptive secant algorithm.

This is due to the use of conflicting *bad* secant directions at different dimensions d to update the adaptive secant algorithm. As a result, we observe oscillations between different admissible dimensions d as a function of the iteration number of the adaptive secant algorithm, as illustrated in Figure 5.7. The reason for such oscillations is that the user supplied weighting parameter α may depend on the current dimension d, and needs adjustment depending on the number of bad secants in each dimension. The same argument is valid for k^* which may be chosen larger for large dimensions.

Algorithm 5.1 Adaptive Secant Algorithm (ASA)

Input: Unit secants K, tolerance k^*, weighting constant α

Output: $\mathbb{P}$ and d such that $\mathbb{P}_d\hat{k} \geq k^*$

1: $K = USV^T$ {Initial basis V}
2: **while** $k_{\min} \ll k^*$ **do**
3: $\min_d ||\mathbb{P}_d\hat{k}|| \geq k^* \quad \forall \hat{k} \in \Sigma$
4: $\mathbb{P} = V_d V_d^T$
5: $S = \{\hat{k} \in \Sigma | ||\mathbb{P}_{d-1}\hat{k} < k^*\}$
6: Update secant covariance matrix Θ' (5.6)
7: $\Theta' = USV^T$
8: **end while**

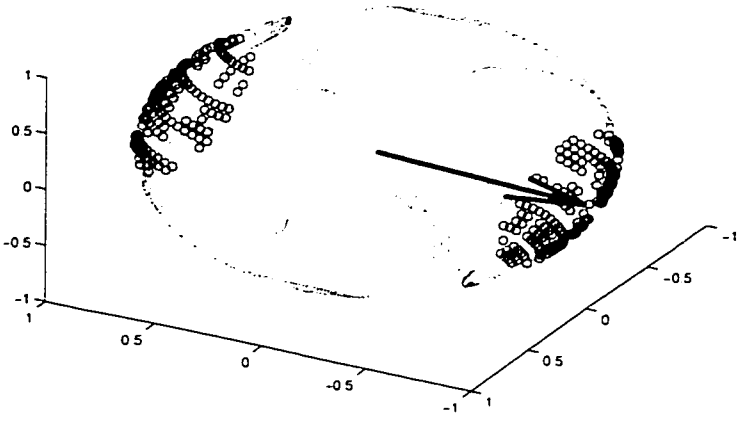


Figure 5.8: The initial direction, v(3) with associated bad secants (o).

5.2.3 Sequential Adaptive Secant Algorithm

An alternative update is proposed here that rests more directly on the constructive proof of the Whitney Embedding Theorem [33]. An admissible projection $\mathbb{P}_d$ is extracted at dimension $d+1$ satisfying $||\mathbb{P}_d\hat{k}|| \geq k^*$, such that $\mathbb{P}_d : \mathbb{R}^{d+1} \rightarrow \mathbb{R}^d$, followed by the immediate application of $\mathbb{P}_d x$, $x \in \mathbb{R}^{d+1}$. We now denote by $\mathbb{P}_{1,d}$ the projector of rank 1, from $\mathbb{R}^{d+1} \rightarrow \mathbb{R}^d$

$$\mathbb{P}_{1,d} = (I - v_{d+1}v_{d+1}^T). \tag{5.8}$$

If the current basis is given as $V_d = [v_1|\ldots|v_{d+1}]$, the projector is computed by $\mathbb{P} = [v_1|\ldots|v_d][v_1|\ldots|v_d]^T$ ($\mathbb{Q} = v_{d+1}v_{d+1}^T$), resulting in a sequence of $q - r$ projections

$$\mathbb{P} = \mathbb{P}_{1,r} \cdots \mathbb{P}_{1,q-1}. \tag{5.9}$$

This new procedure starts naturally with $d + 1 = q$, applying the adaptive secant algorithm to reduce the dimension by one at a time (r=1 in Equation (5.7)), proceeding until no further projection satisfying condition (5.5) can be achieved. The sequential projection algorithm 5.2 is distinguished from Algorithm 5.1, where

Algorithm 5.2 Sequential Adaptive Secant Algorithm (S-ASA)

Input: K, possibly multiple values for k^*, α

Output: $\mathbb{P}$ and d such that $\mathbb{P}\hat{k} \geq k^*$

1: $K = USV^T$ {Initial basis V}
2: $d_{\min} = \min_d ||P_d\hat{k}|| \geq k^* \forall \hat{k} \in \Sigma$
3: $d = d_{\min}$
4: **while** $k_{\min} \ll k^*$ **do**
5: **while** $S \neq \{\emptyset\}$ **do**
6: $S = \{\hat{k} \in \Sigma | ||P_{d-1}\hat{k} < k^*\}$
7: Update secant covariance matrix Θ' (5.6)
8: $\Theta' = USV^T$
9: **end while**
10: $\mathbb{P} = \mathbb{P}\mathbb{P}_{d-1}$
11: Compute new K and V for $\mathbb{P}$
12: **end while**

the dimension is reduced by r in one single projection, after the algorithm has terminated.

Algorithm 5.2 will provide a more controlled way to find a small dimension d, due to the adjustability of the parameters α and k^* at each dimension reduction step. Furthermore, once a projection has been found, the data is immediately projected, prohibiting the effect of oscillations between dimensions. The rank-d adaptive secant algorithm lacks this feature, due to its sensitivity to the bad secant distribution (see Figure 5.7). The application of 5.2 will be demonstrated in an example at the end of this chapter.

5.3 Filtering Secants

An alternative procedure for finding a good projection is based on the idea proposed in [12] of filtering out small secants to produce an admissible projection in the case of noisy observations. The goal here is to successfully identify and filter secants that are dominating the geometric structure of the manifold $\mathcal{M}$. This identification may be performed by investigating the distribution of the secant lengths $||k||$. Figure 5.9 shows such a distribution for the pringle example. The

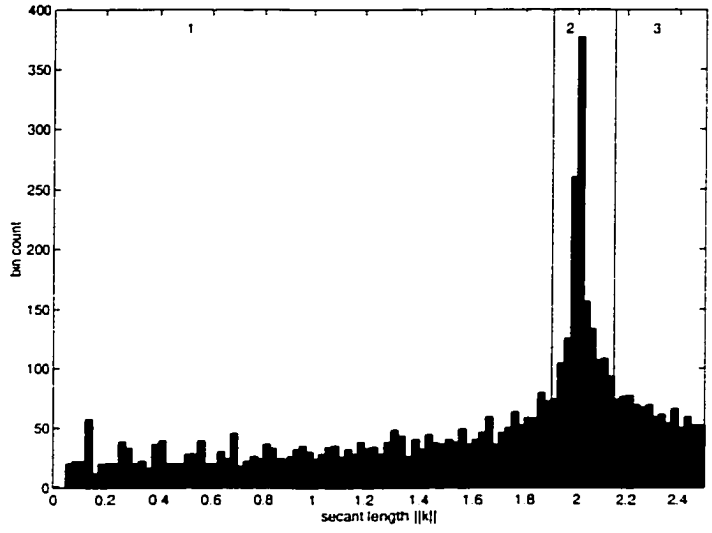


Figure 5.9: The histogram of the lengths of secants $||k||$ for the pringle data. The distribution suggests a separation into three regions.

dominating length occurs at $||k|| \approx 2$. In Figure 5.10 the unit secants and their lengths $||k||$ are illustrated, visualizing their position on the unit-sphere in $\mathbb{R}^3$. The histogram in Figure 5.9 suggests that the secant length might be separated into roughly 3 regions, labeled 1 for small secants, 2 for dominant secants and 3 indicating large secants. Utilizing Figures 5.10 and 5.9 we can conclude that secants in region 1 define the border of the unit secant set Σ, whereas secants that fall in region 2 lie around the equator of the unit sphere in $\mathbb{R}^3$: the large secants fill the space in between. A simple secant length filter would only retain secants with a dominant length at $||k|| \approx 2$ resulting in a secant matrix K_2. After computation of the principal directions of K_2, illustrated in Figure 5.11, the good projection, along $v(3)$ corresponding to the principal direction with the smallest eigenvalue is an appropriate initialization for the adaptive secant algorithm, due to the small number of bad secants associated with this direction.

A more challenging example for finding a good direction is the "Peaks" data, introduced in Chapter 2 in connection with the Autocorrelation Feedback RBF Network, see Equation 3.13. Here we use a domain randomly sampled in $[-2, 2] \times$

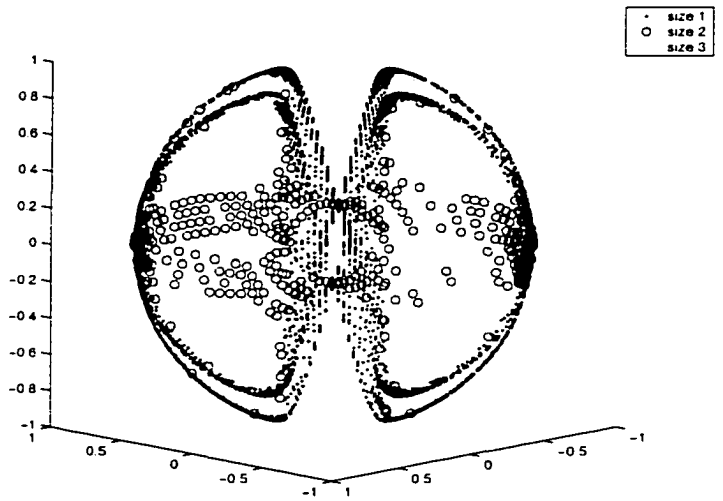


Figure 5.10: Unit secants of the pringle data, now marked according to their length $||k||$.

$[-2, 2]$, as shown in Figure 5.16. This data set requires us to choose $k^* \leq 0.01$ for the successful projection onto the domain spanned by the $x - y$ plane, indicating the ill-conditioning (steepness) of the inverse mapping. The unit secants almost cover the entire unit sphere in $\mathbb{R}^n$, except a relatively small region on the north and south pole. The distribution of lengths of $||k||$ is shown in Figure 5.12; no obvious regions are identifiable, however it was observed that all secants with $||k|| > 4$ lie close to north or south pole - the only direction that permits an admissible projection. All secants $||k|| < 4$ are distributed on the sphere and no obvious filtering can be identified. The adaptive secant algorithm has the problem that basically every direction contains a considerable number of bad secants, which makes the convergence of both Algorithms 5.1 and 5.2 problematic. Here a local secant approach discussed in the following section was successful in finding an admissible projection.

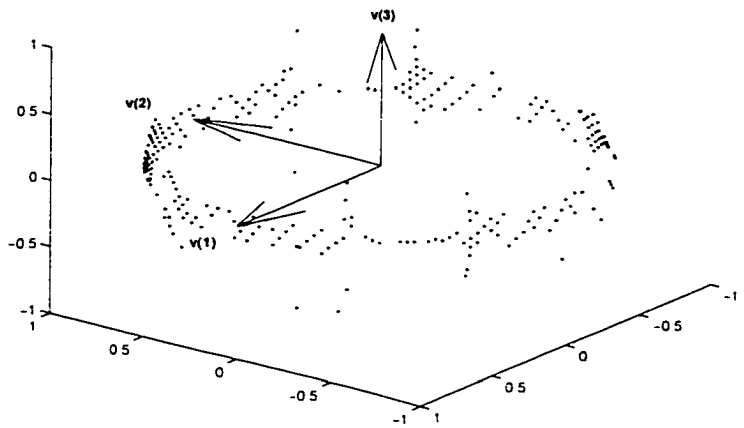


Figure 5.11: The result of keeping secants with $||k|| \approx 2$. The new Secant-SVD basis as indicated, produces a singular vector $v(3)$, associated with the smallest singular value, pointing into the "good" direction.

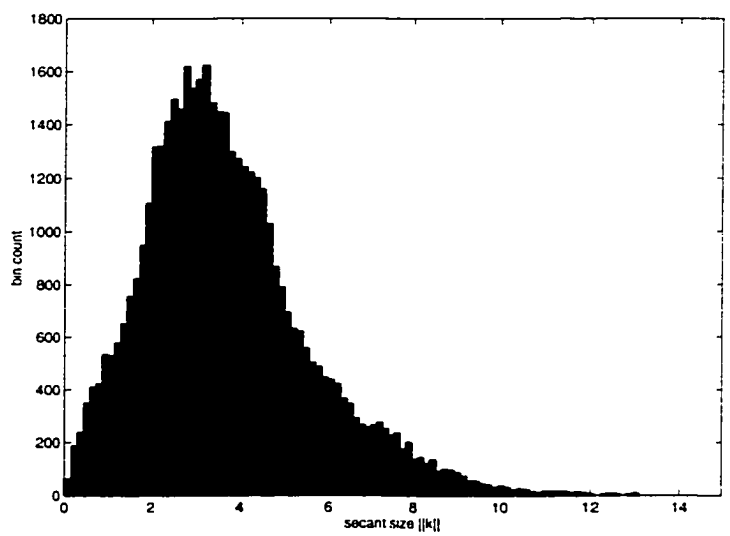


Figure 5.12: The histogram of the lengths of secants for the "Peaks" data.

5.4 The Local Secant Algorithm

The computation of locally admissible projections is motivated by the observation that a global projection should implement a projection that is in some sense common with all local projections. In order to establish a notion of locality, a partition

$$V = \bigcup_{i}^{M} V_i$$

of the data set $\mathcal{A}$, representing the sampled manifold $\mathcal{M}$, is constructed. This partition may be the result of a vector quantization or clustering algorithm, where the cells $\{V_i\}$ are associated with Voronoi regions. As described in [35], an appropriate local representation of the manifold $\mathcal{M}$ may be accomplished using a clustering method which takes (approximated) derivative information into account. Such a technique was developed in the context of modeling dynamical systems via neural charts [35], where a local region V_i is projected onto its tangent space, constructed from a local parameterization of the manifold as the graph $(y, f(y))$. The tangent space is then the span of the Jacobian of f at the cluster center. Consequently, the set of all small, local secants Σ_i lie in that tangent space, which spans the range of a locally admissible projection and can be found using the Secant-SVD algorithm. This locally admissible projection will be denoted by $\pi_i : \mathbb{R}^{d+1} \rightarrow \mathbb{R}^d$, reducing the dimension by one. For the construction of a global projection $\mathbb{P}_d$ we proceed as follows. The admissible directions from each region V_i are collected in the set $\mathcal{V}_d = [v_d^{(1)}| \ldots |v_d^{(M)}]$. The set $\mathcal{V}_2$ is shown in Figure 5.13, where from each center c_i a vector emerges, pointing in the admissible direction. Figure 5.14 (b) shows the set $\mathcal{V}_{d=2}$ in the $x - y$ plane; the argument for a global projection is now that most directions have a counter-direction pointing the opposite way, and the direction with most asymmetry (z-direction) remains as the one resulting admissible direction. Computationally, this direction is associated with the smallest eigenvalue

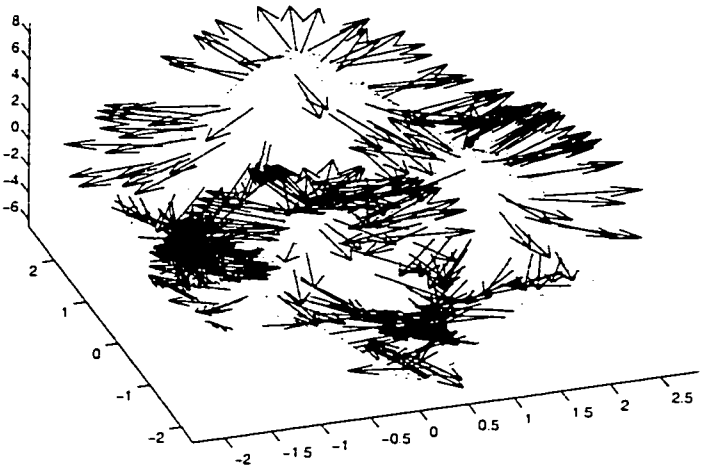


Figure 5.13: Each partition, represented at the origin of the locally admissible projection for the "Peaks" data set.

of the matrix $\Theta_{\mathcal{V}} = \mathcal{V}^T\mathcal{V}$ and is shown in Figure 5.15 (a) and (b), together with the unit secant distribution of the "Peaks" data. Note the relatively small empty region in the z-directional view (b). Figure 5.16 shows $\mathcal{M}$, the sampled "Peaks" manifold , as the graph of the $g : \mathbb{P}\mathcal{M} \subset \mathbb{R}^2 \rightarrow \mathbb{R}$

$$\mathcal{M} = \{(x, g(x)) | x \in \mathbb{P}\mathcal{M} \subset \mathbb{R}^2\},$$

and approximates the original data acceptably well considering the conditioning of the mapping problem.

In an additional experiment, an arbitrarily orthogonal rotation in $\mathbb{R}^3$ was applied to the "Peaks" data such that the z-direction was not longer associated with an good projection direction. Also in this case the local secant algorithm was able to extract the admissible direction.

5.5 Example: Medical Data

A challenging application of intelligent data analysis is the development of computational tools in the field of biomedical signal processing. It is envisioned in

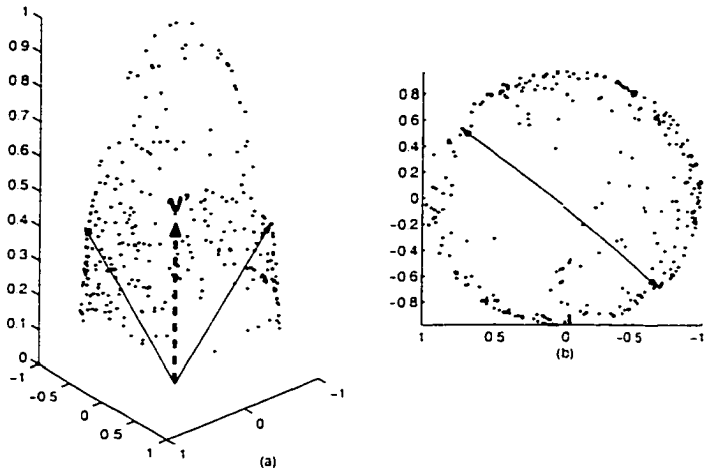


Figure 5.14: (a) The set $\mathcal{V}_2$ of locally admissible directions; (b) the same set viewed from the z-direction. Two locally admissible directions are drawn, the resulting common direction v', extracted using PCA, points into the "good" direction.

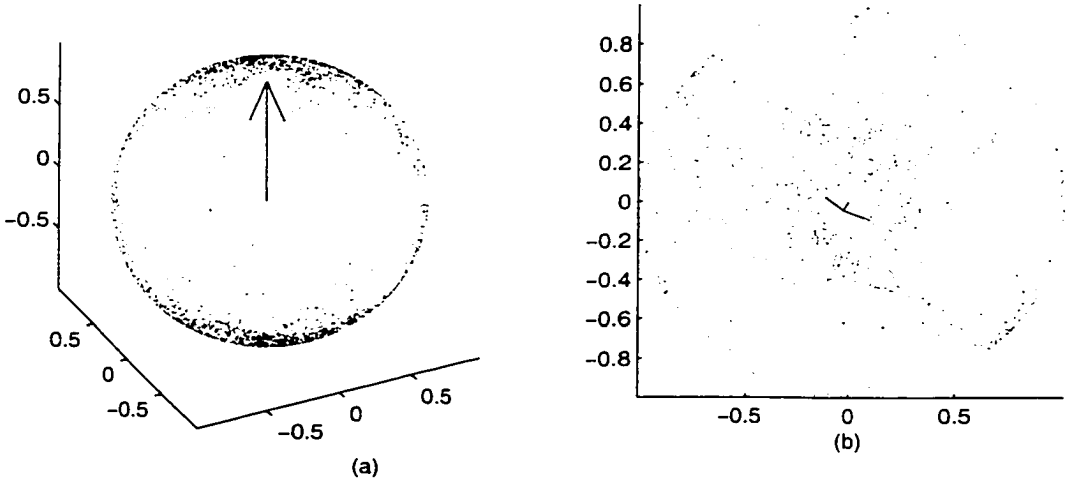


Figure 5.15: The unit secant set for the "Peaks" data and the global projection direction, extracted from the local set of secants.

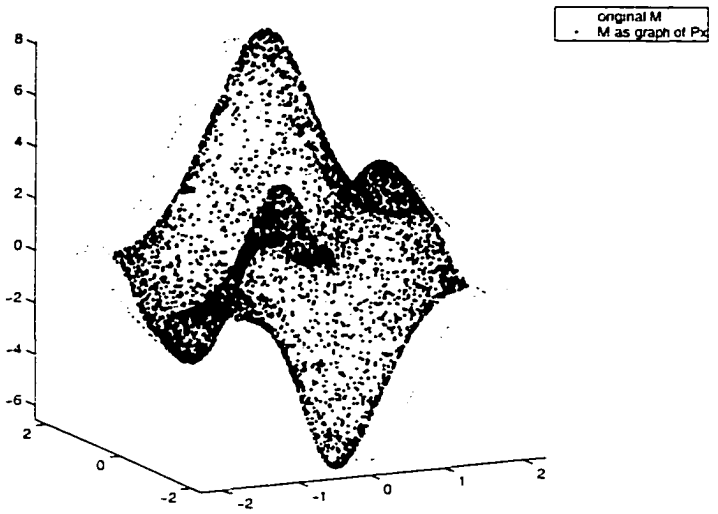


Figure 5.16: The original "Peaks" data (M), compared to the graph of $(x, g(x))$, see Equation 5.4.

the future to create an artificial assistant to help diagnose and propose methods of intervention based on the analysis of large amounts of numerical data that serve to quantify the patient history. To encourage the development of new tools for analyzing complex biomedical signals, the MIMIC (Multi-parameter Intelligent Monitoring for Intensive Care) [56] database at PhysioNet [29] was created. It contains 24 and 48 hours of continuous data recorded from patient monitors in the medical, surgical, and cardiac intensive care units of Boston's Beth Israel Hospital, each signal sampled at 125 Hz (every 0.008 seconds). The study of the changes in one or several physiological variables will often illuminate the underlying physiology, and may be helpful in detecting and separating true patient alerts from monitoring alerts, which can be labeled as false alerts, where no medical intervention is required (about 50% of all alarms are false alarms).

A sample of the data[2] used here is shown in Figure 5.17. It contains eight continuously recorded vital measurements which were mean removed and scaled

[2]Available at http://www.physionet.org/physiobank/database/mimicdb/, patient record 474.

Weighting constant α	100				
Secant norm tolerance k^*	0.2				0.15
Dimension	7	6	5	4	3
Minimum Secant Norm	0.602	0.500	0.487	0.320	0.126
Number of iteration	3	11	14	40	18
Total number of iterations	86				

Table 5.1: Sequential adaptive secant results for the MIMI data. Compare to Figure 5.18.

to unit variance in a preprocessing step. The task here is to find an appropriate projection to a low dimensional space, optimally to $\mathbb{R}^3$ for visualization purposes, and to generate a low dimensional data parameterization for the first step in the application of the WRN to the data modeling problem. Figure 5.18 compares minimum secant norms for the Secant-SVD basis, the data PCA data basis and the sequential adaptive secant algorithm (S-ASA). The details of the application of the S-ASA algorithm, such as number of iterations for each dimension, α and k^* are summarized in Table 5.1. Figure 5.19 illustrates the performance of the S-ASA compared with the adaptive secant algorithm where $k^* = 0.2$ and $\alpha = 100$ for 100 iterations (different parameter settings were tested for the adaptive secant algorithm, with no significant improvement in the minimum secant norm). Note that for $d = 7$ the adapted secant basis is worse that the unadapted secant basis, because the algorithm is not focused on $k \approx 0.4$. Finally Figure 5.20 visualizes the application of the sequence of projections to the MIMI data set.

5.6 Recipe and Summary

Several new procedures have been introduced here for the task of finding good quality projections, in the sense that the minimum secant norm is as large as possible whereas the dimension of the range of $\mathbb{P}$ is as small as possible. To this end, all algorithms need considerable interactions with the user to find a good set of

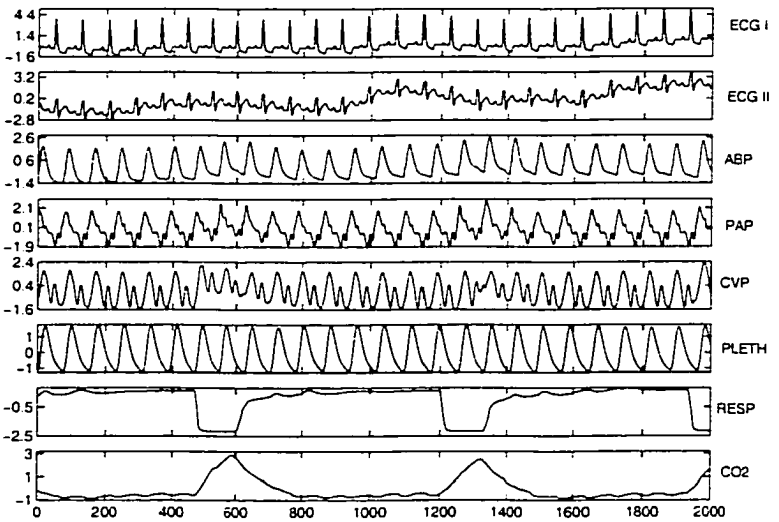


Figure 5.17: ABP (arterial blood pressure), PAP(pulmonary arterial pressure), CVP (central venous pressure), PLETH (fingertip plethysmograph), RESP (respatory rate) and CO2 (CO2 level).

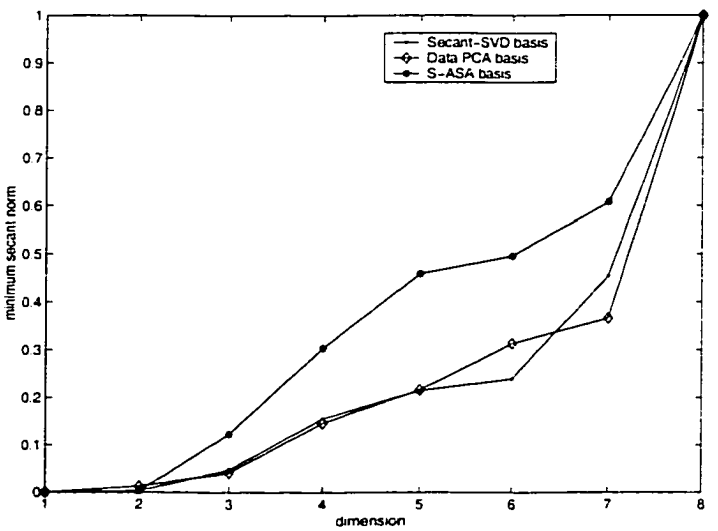


Figure 5.18: The minimum norms of the projected unit secants of the MIMI data as a function of dimension for the Secant-SVD basis, the PCA data basis and the sequential adaptive secant algorithm.

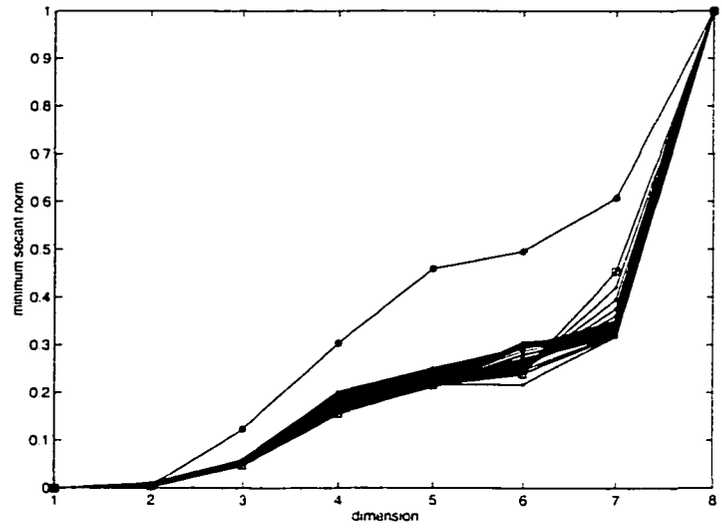


Figure 5.19: The minimum norms of the projected unit secants of the MIMI data as a function of dimension compares the performance of the adaptive secant algorithm $(-\cdot-)$ and the sequential adaptive secant algorithm $(-*-)$. The initial SVD-basis is marked as $(-\square-)$.

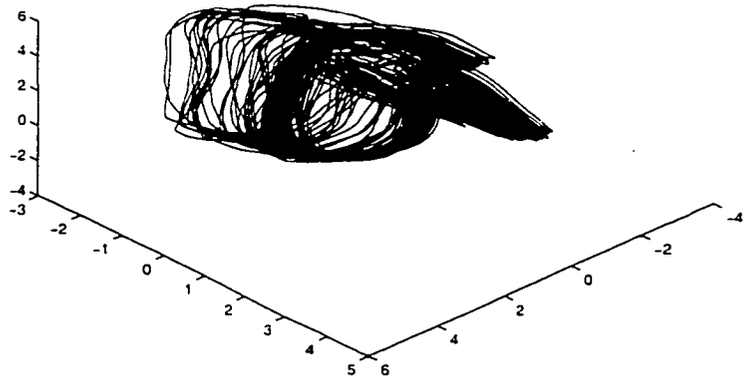


Figure 5.20: The visualization of the projected MIMI data into $\mathbb{R}^3$.

adaption parameters. However, the sequential adaptive secant algorithm approach is more user-friendly in the sense that each projection in a sequence of projections may be optimized individually, which also gives superior results compared with the adaptive secant algorithm. In the following a summary of the data secant analysis is proposed to assist with the quest for a good projection:

1) Scaling: The analysis of raw observational data often demands an appropriate scaling method. Recorded signals often come with their own, inherent scale. If many of them are combined to form a multivariate signal, or data matrix, further analysis generally requires scaling of the data which can be difficult without prior knowledge (e.g. noise levels) about the data characteristics. Also, the performance of PCA and projections based on secants are very sensitive to different scaling methods of the data. Nevertheless, useful scaling methods include mean removal combined with the standardization of the variance in each mode to one. The Min/Max scaling usually scales the signals to be between 1 and -1. Using PCA, a whitening transformation of the data may be performed, which is also essentially the first preprocessing step in applying independent component analysis. For the application of maximum noise fraction scaling see Chapter 3.

2) Change of basis: Statistical second order information about the covariance structure of the data can be extracted by performing a PCA analysis of the data. Note that resulting data PCA basis may not provide a good projection, since it is not optimized for that purpose, but can be employed in a comparison, once a basis for a good projection is found. Higher order statistics can be extracted by performing an independent component analysis (ICA) to further enhance the understanding of the data; also this basis may be employed and tested for good projections, but again it is not optimized to produce good projections.

3) Secant Analysis: The computation and visualization of the secants and unit secant set is the essential tool for finding good projections:

   (a) Secant-SVD: The PCA basis of the unit secant matrix may achieve the construction of a basis for a good projection. This is particularly true if the secant length distribution can be employed to filter dominant secants. Furthermore, this basis is useful to initialize iterative algorithms for optimizing the minimum secant norm.

   (b) S-ASA: An improved basis may be found by iteratively weighing bad secants and terminate with an acceptable k_{min}, with the possibility of adjusting the parameters to optimize the conditioning of the projection at each dimension reduction step.

   (c) Local Secants: The concept of obtaining a global admissible projection from local projections is particularly attractive if a useful partition of the data is available, each resulting in a significantly different local projection.

Chapter 6

CONCLUSION AND OUTLOOK

We conclude this thesis with a discussion and an outlook of the approaches and algorithms that were applied in the process of modeling noisy data observations. The central question to answer is, if we accomplished the task of construction a *model* that

... generalizes well to the data,

... recovers the hidden states and the intrinsic dimension of the observations,

... provides us with a method of dealing with noisy measurements,

... is based empirically on successful and efficient algorithms.

This thesis uses the methodology of dimension reduction mappings to recover hidden states in "real-world" data, possibly contaminated with a noise process. In our first approach, these hidden states where extracted using a clustering algorithm. This was successful due to the observation that hidden states often occupy different regions in high dimensional space, considerably separated with respect to some similarity measure. The concept of a weighted bi-directional Hebb rule provides a flexible tool to build a connectivity structure to discover these regions. Otherwise, this may only be accomplished with a large amount of cluster centers either

in the context of topology preserving networks under the condition of dense centers[1], or some other measure of low quantization error, which possibly leads to bad generalization in a noisy environment and requires large computational power to process the data. Note that "real-world" observations often exhibit missing values, in the sense that proper class labels, e.g., for normal operation or some failure, are not supplied, making the application of traditional classification procedures for detecting the associated states difficult. Furthermore, the notion of locality via second order Voronoi cells leads to an efficiency improvement compared to many vector quantization algorithms, suffering from the evaluation of global similarity measures. An attractive implementational feature is the flexibility of the proposed algorithmic additions, simplifying their combination with traditional clustering methods due to their modular character of, e.g., the weighted bi-directional Hebb rule, or the growing component in G-LLBG. Furthermore, we showed the successful combination of a neural motivated learning rule (Hebb rule) with the popular batch LBG and k-means algorithms. In the future we anticipate to add intelligent computational structures, e.g., $k-d$ trees [66], to our algorithm.

The autocorrelation feedback mechanism introduced here for radial basis functions, provides a method to extract a nonlinear relationship between a domain data set and a noisy target. This task occurs for example in the reconstruction step in the WRN. This novel approach for solving the bias - variance dilemma [28] rests on the correlation structure of the approximated residuals, using the prior information that their anticipated behavior is iid. The combination with a local optimization procedure leads to a learning algorithm that eliminates most *ad hoc* parameters, simplifying the model building process. The discovery of hidden states

---

[1]Note that the definition of dense centers in [52] does not hold if the proper mathematical definition of a manifold is used.

in a function approximation problem can be associated with the identification of basis functions that match certain parts of the target, which is only possible if a parsimonious representation of the domain-target map is achieved. The ACF-RAN contributes to this interpretation due to the locality of the training process.

In the context of the WRN, the set of secants naturally, via their definition, contain information about the geometric structure of the data, e.g., their length and distribution on the unit sphere. In this thesis they are utilized in the most difficult step in the WRN, that is the construction of a good quality projection in order to parameterize the data. This difficulty is due to the balancing between a low dimensional representation and a well-conditioned inverse. This point should not be viewed as a setback to the automation of the algorithm, especially in view of the drawbacks associated with alternative approaches to the nonlinear dimension reduction problem, such as the bottleneck network [41]. The advantage lies in the controllability of the balance between d and k_{min}. A low d might be inappropriate if the inverse is too ill-conditioned, prohibiting good generalization. In order to approach this trade-off from a practical point of view, we introduce new algorithms that utilize the properties of the secant set to give us insight into the geometric structure of the object under investigation, including a version that explores the construction of local admissible projections, successfully providing us with a lower dimensional representation of the data, while allowing the construction of a well-conditioned inverse. Expanding this argument, we will be able to judge alternative dimension reduction algorithms in terms of k_{min} and d and utilize secants to quantify their differences.

In the geometric approach of data modeling the maximum noise fraction approach is a very natural one, since it relies on the geometric separation of a noise and signal subspace. That this is accomplished by the investigation of the (temporal) correlation structure is an interesting twist to the stationary, geometric

approach: the signal geometrically separates from the noise process due to a difference in covariance structure (2nd order statistics) of the temporal shifted signal. Furthermore, scaling with respect to an estimation of the noise is a more natural (and practical) approach to data preprocessing, compared to principal component analysis, which highly depends on the applied scaling method. Even in a noise-free context, where differencing can be viewed as an estimate of the derivative of the signal, an ordering of the MNF eigenvectors in terms of their smoothness is provided: the application of MNF transformation can be viewed as the application of a smoothing filter. Finally, the integration of MNF into the WRN architecture provides us with a principled approach to data parameterization and reconstruction in a noisy environment.

Future work in the area of geometric data modeling will focus on the connection with a probabilistic description of the data. It is envisioned to utilize geometric based transformations to enhance our understanding of methods based on statistical data analysis. That this is a fruitful approach was already demonstrated in the discovery of striking similarities between the maximum noise fraction transform and the independent component analysis [3]. Furthermore, the construction of good projectors and the extraction of *interesting* directions in a probabilistic sense (see, e.g., Projection Pursuit [23]) seem to have a common architecture, despite the objective function a projection is optimizing are quite different. The statistical investigation of relations between hidden variables, e.g., in terms of their independence, would give further insight into the geometric interpretation of the observed data.

In the context of clustering, a current task lies in the development of efficient on-line algorithms that deal with data arriving in streams. Also, resource allocating networks based on residual and input correlations will deal in the future with data arriving sequentially.

Bibliography

[1] M.R. Allen and L.A. Smith. Optimal filtering in singular spectrum analysis. *Physics Letters A*, 234:419–428, 1997.

[2] M. Anderle and M. Kirby. Adaptive clustering based on local neighborhood interactions. In *Proceedings of SPIE*, volume 3807 of *Advanced Signal Processing Algorithms, Architecture, and Implementations*, pages 288–297, 1999.

[3] M. Anderle and M. Kirby. Filtering noisy time series: Keeping the baby and most of the bathwater. In *Conference Digest, Fifth IMA International Conference on Mathematics in Signal Processing, Univeristy of Warwick*, 2000.

[4] C. Anderson. Q-learning with hidden-unit restarting. In S.J. Hanson, J.D. Cowan, and C.L. Giles, editors, *Advances in Neural Information Processing Systems 5*, pages 81–88. Morgen Kaufmann Publishers, San Mateo, 1993.

[5] P. Baldi and K. Hornik. Neural networks and principal component analysis: Learning from examples without local minima. *Neural Networks*, 2:53–58, 1989.

[6] A. Belouchrani, K. Abed-Meraim, J.-F. Cardoso, and E. Moulines. A blind soure separation technique using second order statistics. *IEEE Transacions on Signal Processing*, 45:434–444, 1997.

[7] D.A. Belsley. *Conditioning Diagnostics: Collinearity and Weak Data in Regression.* New York: John Wiley and Sons, 1991.

[8] S.A. Billings and W.S.F Voon. Correlation based model validity tests for non-linear models. *International Journal of Control*, 44:235–244, 1986.

[9] C.M. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, Oxford, U.K., 1995.

[10] P.J. Brockwell and R. A. Davis. *Time Series: Theory amd Methods.* Springer-Verlag, New York, 2 edition, 1988.

[11] D.S. Broomhead and D. Lowe. Multivariable functional interpolation and adaptive networks. *Complex Systems*, 2:321–355, 1988.

[12] D.S. Broomhead and M.Kirby. The whitney reduction network: a method for computing autoassociative graphs. *accepted for publication by Neural Computation*, 1998.

[13] D.S. Broomhead and M.Kirby. A new approach for dimensionality reduction: Theory and algorithms. *SIAM Journal of Applied Mathematics*, 60(6):2114–2142, 2000.

[14] J. Cardoso and A. Souloumiac. Blind beam forming for non-gaussian signals. *IEE Proceedings F*, 140(6):771–774, 1993.

[15] J. Case. Data mining emerges as a new discipline in a world of increasingly massive data sets. *SIAM News*, 32(10), 1999.

[16] S. Chen, C.F.N Cowan, and P.M. Grant. Orthogonal least squares learning algorithm for radial basis function networks. *IEEE Transactions on Neural Networks*, 9:1597–1617, 1991.

[17] P. Comon. Independent component analysis, a new concept? *Signal Processing*, 36:287–314, 1994.

[18] T.H. Cormen, C.E. Leiseron, and R.L. Rivest. *Introduction to Algorithms.* MIT Press, 1998.

[19] T.M. Cover and J.A. Thomas. *Elements of Information Theory.* Wiley Series in Telecommunications. John Wiley and Sons, 1991.

[20] G. Deco and D. Obradovic. *An Information-Theoretic approach to neural computing.* Perspectives in Neural Computing. Springer, 1996.

[21] R.O Duda and P.E. Hart. *Pattern Classification and Scene Analysis.* John Wiley and Sons, 1973.

[22] J.H. Friedman and W. Stuetzle. Projection pursuit regression. *Journal of the American Statistical Association*, 76:817–823, 1981.

[23] J.H. Friedman and J.W. Tukey. A projection pursuit algorithm for exploratory data analysis. *IEEE Transactions on Computers*, C-23:881–889, 1974.

[24] B. Fritzke. Growing cell structures-a self organizing network for unsupervised and supervised learning. *Neural Networks*, 7(9):1441–1460, 1994.

[25] B. Fritzke. A growing neural gas network learns topologies. In D.S. Touretzky G. Tesauro, editor, *Advances in Neural Information Processing Systems Vol. 7*, pages 625–632. MIT Press, Cambridge MA, 1995.

[26] B. Fritzke. The lbg-u method for vector quantization - an improvement over lbg inspired from neural networks. *Neural Processing Letters*, 5:35–45, 1997.

[27] K. Fukunaga. *Introduction to Statistical Pattern Recognition.* Morgan Kaufann, 2 edition, 1990.

[28] S. Geman, E. Bienenstock, and R. Doursat. Neural networks and the bias/variance dilemma. *Neural Computation*, 4:1–58, 1992.

[29] A.L. Goldberger, L.A.N. Amaral, L. Glass, J.M. Hausdorff, P.C. Ivanov, R.G. Mark, J.E. Mietus, G.B. Moody, C.K. Peng, and H.E. Stanley. Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220, June 2000. [Circulation Electronic Pages;http://circ.ahajournals.org/cgi/content/full/101/23/e215].

[30] G.H. Golub and C.F. van Loan. *Matrix Computations.* John Hopkins University Press, 1996.

[31] A.A. Green, M. Berman, P. Switzer, and M.D. Craig. A transformation for ordering multispectral data in terms of image quality with implications for noise removal. *IEEE Transcations on Geoscience and Remote Sensing*, 26:65–74, 1988.

[32] J. Hertz, A. Krogh, and R.G. Palmer. *Introduction to the Theory of Neural Computation.* Addison-Wesley, 1991.

[33] M.W. Hirsch. *Differential Topology.* Grad. Texts in Math. 33. Springer, 1976.

[34] P. J. Huber. Projection pursuit. *Annals of Statistics*, 13:435–475, 1985.

[35] D. Hundley. *Local Nonlinear Modeling Via Neural Charts.* PhD thesis, Colorado State University, Dept. of Mathematics, 1998.

[36] A. Hyvärinen and E. Oja. A fast fixed-point algorithm for independent component analysis. *Neural Computation*, 9:1483–1492, 1997.

[37] F. Jia, E.B. Martin, and A.J. Morris. Non-linear principal component analysis with application to process fault detection. *International Journal of Systems Science*, 31(11):1473–1487, 2000.

[38] V. Kadirkamanathan and M. Niranjan. A function estimation approach to sequential learning with neural networks. *Neural Computation*, 5:954–975, 1993.

[39] M. Kirby. *Geometric Data Analysis.* John Wiley and Sons, New York, 1 edition, 2001.

[40] T. Kohonen. *Self-organization and associative memory.* Springer Series in Information Sciences 8. Springer, 1988.

[41] M.A. Kramer. Nonlinear principal component aanalysis using autoassociative neural networks. *AIChE Journal*, 37(2):233–243, 1991.

[42] D. Lee and S. Baek K. Sung. Modified k-means algorithm for vector quantizer design. *IEEE Signal Processing Letters*, 4:2–4, 1997.

[43] J.B Lee, A.S Woodyatt, and M. Bergman. Enhancement of high spectral resolution remote sensing data by noise-adjusted principal components transform. *IEEE Transcations on Geoscience and Remote Sensing*, 28:295–304, 1990.

[44] A. Leonardis and H. Bischof. An efficient mdl-based construction of rbf networks. *Neural Networks*, pages 963–973, 1998.

[45] Y. Linde, A. Gray, and R.M. Gray. An algorithm for vector quantizer design. *Proceedings of the IEEE*, 28(1):84–95, January 1980.

[46] L. Ljung. *System Identification - Theory for the User*. Information and System Science. Prentice Hall, Inc., 2 edition, 1999.

[47] D. Lowe and A. McLachlan. Modelling of non-stationary processes using radial basis function networks. In *Proceedings of the 4th IEE International Conference on Artificial Neural Networks*, pages 300–305. IEE Conference Publications, 1995.

[48] D. Lowe and A.R. Webb. A hybrid optimisation strategy for adaptive feedforward layered. Technical Report RSRE Memorandum 4193, Royal Signals and Radar Establishment, 1988.

[49] D.G. Luenberger. *Linear and Nonlinear Programming*. Addison-Wesley, Menlo Park, California, 1984.

[50] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical statistics and probability*, pages 281–297. University of California Press, 1967.

[51] T. Martinez, S.Berkovich, and Klaus Schulten. Neural-gas network for vector quantization and its application to time-series prediction. *IEEE Transactions on Neural Netwoks*, 4(4):558–569, 1993.

[52] T. Martinez and K. Schulten. Topology representing networks. *Neural Networks*, 3:507–522, 1994.

[53] A. McLachlan. An improved novelty criterion for resource allocating networks. Technical Report NCRG/96/023, Neural Computing Research Group, Aston Univerity, UK, 1996.

[54] A. McLachlan and D. Lowe. Tracking of non-stationary time-series using resource allocating rbf networks. In R. Trappl, editor, *Cybernetics and Systems '96: Proceedings of the 13th European Meeting on Cybernetics and Systems Research*, pages 1066–1071, 1996.

[55] M. McLoone, M.D. Brown, and G. Irwin. A hybrid linear/nonlinear training algorithm for feedforward neural networks. *IEEE Transactions on Neural Networks*, 9:669–683, 1998.

[56] G.B. Moody and R.G. Mark. A database to support development and evaluation of intelligent intensive care monitoring. *Computers in Cardiology*, pages 657–660, 1996.

[57] J. Moody and C. Darken. Fast learning in networks of locally-tuned processing units. *Neural Computation*, 1:281–294, 1989.

[58] B. De Moor, J. Vandewalle, and J. Staar. Oriented energy and oriented signal-to-signal concepts in the analysis of vector sequences and time series. In E. Deprettere, editor, *SVD and Signal Processing: Algorithms, Applications and Architectures*, pages 209–232. North Holland, 1987.

[59] A. W. Moore. Very fast em-based mixture model clustering using multiresolution kd-trees. *Advances in Neural Information Processing Systems*, 11, 1998.

[60] A. W. Moore. The anchors hierachy: Using the triangle inequality to survive high dimensional data. In *UAI-2000: The Sixteenth Conference on Uncertainty in Artificial Intelligence*, 2000.

[61] I.T. Nabney, A. McLachlan, and D. Lowe. Practical methods of tracking non-stationary time series applied to real world data. In S. K. Rogers and D. W. Ruck, editors, *AeroSense '96 : Applications and Science of Artificial Neural Networks II.SPIE Proceedings No. 2760.*, pages 152–163, 1996.

[62] A. Nag and J. Ghosh. Flexible resource allocating network for noisy data. In *SPIE Conference on Applications and Science of Computational Intelligence*, volume 3390, pages 551–559, 1998.

[63] M.J.L. Orr. Regularization on the selection of radial basis function centers. *Neural Computation*, 7:606–623, 1995.

[64] J. Platt. A resource allocating network for function approximation. *Neural Computation*, 3:213–225, 1991.

[65] M.J.D. Powell. The theory of radial basis functions in 1990. In W. Light, editor, *Advances in Numerical Analysis II: Wavelets Subdivision and Radial Basis Functions*, pages 105–210. Oxford Univeristy Press, 1992.

[66] F.P. Prepatata and M.I. Shamos. *Computational Geometry.* Springer, 1985.

[67] B.D. Ripley. *Spatial statistics.* Wiley, New York, 1981.

[68] P. Switzer and A. Green. Min/max autocorrelation factors for multivariate spatial imagery. Technical Report 6, Dept. of Statistics, Stanford University, 1984.

[69] Third Conference on Neural Networks and Their Applications, Kule, Poland. *Statistical Control of Growing and Pruning in RBF-like Neural Networks*, October 1997.

[70] P. Veprek and A.B. Bradley. An improved algorithm for vector quantizer design. *IEEE Signal Processing Letters*, 7(9):250–252, 2000.

[71] B. Widrow and M.E. Hoff Jr. Adaptive switching circuits. In *IRE WESCON Convention Record*, pages 96–104, 1960.

[72] L. Yingwei, N. Sundararajan, and P. Saratchandran. A sequential scheme for function approximation using minimal radial basis function neural networks. *Neural Computation*, 9:461–478, 1997.

[73] L. Yingwei, N. Sundararajan, and P. Saratchandran. Performance evaluation of a sequential minimal radial basis function (rbf) neural network learning algorithm. *IEEE Transactions on Neural Networks*, 9:308–318, 1998.

[74] B. Zhang. Generilized k-harmonic means. In *Proceedings of the First SIAM International Conference on Data Mining*, pages 455–467, 2001.

Appendix A

NOISE ADJUSTED PRINCIPAL COMPONENT ANALYSIS

The maximum noise fraction (MNF) decomposition we present here follows a reformulation [43] of the original work [31]. The two alternative approaches related to the simultaneous diagonalization of matrices are discussed in [27].

Let E be the eigenvector matrix of the noise covariance matrix Σ_n, with the usual normalization:

$$E^T \Sigma_n E = \Delta_n, \quad E^T E = I$$

where Δ_n is the diagonal matrix of the eigenvalues of Σ_n. We renormalize by the noise-whitening matrix $F = E\Delta_n^{-1/2}$:

$$F^T \Sigma_n F = I, \quad F^T F = \Delta_n^{-1}$$

Now introducing $z_i = F^{-1}\psi_i$ (the retransformation will be $\psi_i = Fz_i$) where ψ_i is the MNF eigenvector and z_i its representation in the whitened space. With $\Sigma_s = \Sigma - \Sigma_n$ if follows

$$\mathrm{SNR}_i(z_i) = \frac{z_i^T F^T \Sigma_s F z_i}{z_i^T z_i} = \frac{z_i^T F^T (\Sigma - \Sigma_n) F z_i}{z_i^T z_i} \tag{A.1}$$

$$= \frac{z_i^T \Sigma_W z_i}{z_i^T z_i} + 1. \tag{A.2}$$

with noise-adjusted data covariance matrix Σ_W. The principal components of Σ_W maximize the expression for the signal-to-noise ratio (A.2) above, subject to

constraint $z_i^T z_j = \delta_{ij}$. These are the noise-adjusted principal components, corresponding to the maximum noise fraction vectors in reverse order.

$$Z^T \Sigma_W Z = \Lambda_W, \quad Z^T Z = I$$

where $\Lambda_W = \text{diag}\{\lambda_{W,i}\}_{i=1}^q$ is the diagonal matrix of the eigenvalues equal $\text{SNR}_i + 1$. Finally the the desired *noise adjusted principal component* transform (NAPC) can be summarized by

$$H = FZ = E\Delta_N^{-1/2}\Sigma_W,$$

therefore the NAPC can be implemented by a two-stage procedure comprised of a first-stage process using F to whiten the noise, and a second stage using Z to perform a PCA transform, followed by the back-transformation to the non-whitened space.

Appendix B

THE WHITNEY REDUCTION NETWORK IN MAXIMUM NOISE FRACTION SPACE

B.1 Norms

The rank-q representation of the data matrix X using the MNF eigenvectors is given by

$$X_q = \Phi_q \Phi_q^T X,$$

retaining D MNF eigenvectors leads to the rank-D representation of X

$$X_D = \Phi_D \Phi_D^T X.$$

If we chose to represent X_D by using Φ_D instead, we seek an appropriate norm in order to evaluate distances between points $\phi_D^{(i)} \in \mathbb{R}^D$, e.g. to construct an parametrization of $\phi_D^{(i)} \in \mathbb{R}^D$ employing a projection using the concept of secants. From the inner product of X

$$XX^T = \Phi_D B_D B_D^T \Phi^T$$

we can derive an weighted inner product norm such that

$$||x^{(i)}|| = ||\phi_D^{(i)}||_{\Sigma_B},$$

where $\Sigma_B = B_D B_D^T$.

B.2 Secants

We employ the norm Σ_B for the calculation of unit secants between points $\phi^{(i)} \in \mathbb{R}^D$ as

$$\hat{k} = \frac{\phi^{(i)} - \phi^{(j)}}{||\phi^{(i)} - \phi^{(j)}||_{\Sigma_B}}$$

This set of unit secants is now used as discussed in chapter to find a good projection based on a basis V for $\mathbb{R}^D$. We require this basis to be orthogonal with respect to the weighted inner product norm Σ_B

$$V^T \Sigma_B V = I.$$

In addition, we seek a basis V as the eigenvectors of the covariance matrix Θ of the unit secant matrix $\Theta = K^T K$, following the Secant-SVD and the adaptive secant algorithm concept of [13].

$$\Theta \Sigma_B v_i = \lambda_i v_i \tag{B.1}$$

Following the left-multiplication of Equation (B.1) with $v_i^T \Sigma_B$,we compute the eigenvectors v_i as the solution to the symmetric generalized eigenvalue problem

$$v_i^T \Sigma_B \Theta \Sigma_B v_i = \lambda_i v_i^T \Sigma_B v_i,$$

such that the basis V satisfies $V^T \Sigma V = I_q$ and $V^T \Sigma_B \Theta \Sigma_B V = \Lambda$, diagonializing the secant covariance matrix with respect to the weighted inner product norm, where Λ, is a diagonal matrix with λ_i along its diagonal.

The basis $V = [v_1|\ldots|v_d|v_{d+1}|\ldots|v_D] = [V_1|V_2]$ obtained by solving (B.2) via, e.g., Algorithm 8.7.1 in [30], is now employed to parameterize a vector $\phi^{(i)} \in \mathbb{R}^D$

$$\Pi_d^{(i)} = (V_1 \phi^{(i)})_{\Sigma_B} \quad \text{and } \Pi_d^{(i),\perp} = (V_2 \phi^{(i)})_{\Sigma_B},$$

again using the weighted inner product norm to evaluate the scalar products.

Appendix C

NOISE COVARIANCE ESTIMATION

C.1 Temporal Correlations

Using concepts form spatial statistics [67] in a temporal context, and covariance definitions form multivartiate time-series analyis [10] we are able to derive some of the statements given in [68] and [31] for time signals (see also [6]). Let us again consider the observation of a multivariate time signal $x(t) \in \mathbb{R}^q$, sampled P times on $0 \leq t \leq T$

$$x^T(t) = (x_1(t), .., x_q(t)), \quad t = 1 \ldots T,$$

which is composed of a deterministic signal uncorrelated with additive noise at times t ($E[x(t)n(t)] = 0$)

$$x(t) = s(t) + n(t).$$

The covariance of X is given as

$$\Sigma = \mathrm{Cov}[X(t), X(t)] = \langle x(t)x^T(t) \rangle$$

The application of a temporal shift Δ (in an application we choose usually $\Delta = 1$), assuming weak stationarity [10] and uncorrelatedness between signal and noise, leads to the temporal covariance matrices of the signal X

$$\mathrm{Cov}[X(t), X(t+\Delta)] = \Gamma(\Delta) \tag{C.1}$$

$$= \mathrm{Cov}[S(t), S(t+\Delta)] + \mathrm{Cov}[N(t), N(t+\Delta)] \tag{C.2}$$

$$= \Gamma_s(\Delta) + \Gamma_n(\Delta), \tag{C.3}$$

describing the temporal correlation structure of the signal at time shift Δ, where $\Gamma(0) = \Sigma$ and $\Gamma(\Delta)^T = \Gamma(-\Delta)$.[1] A natural estimator of $\Gamma(\Delta)$ is

$$\Gamma_{ij}(\Delta) = \begin{cases} 1/T \sum_{t=1}^{T-\Delta} x_i(t) x_j(t+\Delta) & 0 \leq \Delta \leq T-1 \\ \Gamma_{ij}(-\Delta)^T & -T+1 \leq \Delta < 0 \end{cases}$$

The diagonal entries are the autocovarriance functions for the individual time series, whereas the off-diagonal elements contain the cross-covariances between time-series as a function of Δ. Of interest is the covariance of the shifted or differenced time-series. This shift is referred to as cointegration for multi valued time-series

$$\Sigma_\Delta = \mathrm{Cov}[X(t) - X(t+\Delta), X(t) - X(t+\Delta)]$$

employing the expression for the temporal covariance, and again assuming stationary, Σ_Δ may be written

$$\Sigma_\Delta = 2\Sigma - (\Gamma(\Delta) + \Gamma(-\Delta)).$$

Note that if a signal exhibits no temporal correlation, the covariance of the cointegrated time-series is twice as large as the original time-series. Decomposing Σ_Δ into signal and noise covariances leaves us with

$$\Sigma_\Delta = 2\Sigma - (\Gamma_s(\Delta) + \Gamma_s(\Delta)^T) - (\Gamma_n(\Delta) + \Gamma_n(\Delta)^T) \tag{C.4}$$

In order to make use of the concept of proportional covariances [31], we need to make the following assumptions for small Δ and a constant c

$$\Gamma(\Delta) = \Gamma(-\Delta)) = c\Gamma(0) = c\Sigma \tag{C.5}$$

the application of (C.5) to (C.4), assuming different constants for signal and noise results in

$$\frac{1}{2}\Sigma_\Delta = 2\Sigma - \alpha\Sigma_s - \beta\Sigma_n$$

---

[1]Note that for univariate time-series $\gamma(\Delta) = \gamma(-\Delta)$, however $\Gamma(\Delta)$ is in general not symmetric.

using the decomposition of $\Sigma_s = \Sigma - \Sigma_n$

$$\frac{1}{2}\Sigma_\Delta = \Sigma - \alpha(\Sigma - \Sigma_n) - \beta\Sigma_n \tag{C.6}$$

$$\Sigma_\Delta = 2(1-\alpha)\Sigma + (\alpha - \beta)\Sigma_n \tag{C.7}$$

Under the assumption that the deterministic signal has high temporal correlation at shift Δ, we may choose $\alpha \approx 1$. In addition, the use of proportional covariances assumes that the local temporal correlation at shift Δ is the same in all modes, e.g for $\Delta = 1, \Gamma(1) \approx \Gamma(-1) \approx \Sigma$, this is certainly true only as a function of the sampling rate Ω of the observation $x(t)$. The opposite is assumed for the noise, i.e. low temporal correlation at shift Δ, such that $\beta \approx 0$, like temporal white noise; note that in this case Σ_n is not diagonal compared to $\Gamma_n(\Delta)$. Using (C.7) with $\alpha = 1$ and $\beta = 0$, the estimate of the noise covariance matrix in terms of Σ_Δ becomes

$$\Sigma_n = \frac{1}{2}\Sigma_\Delta$$

and is in general a good approximation if the sampling rate Ω is high enough, an exact relationship is left for further research.

The following is a list of results using the relation (C.7)

- Using Σ_Δ instead of Σ_n in the generalized eigenproblem, results in the same set of generalized eigenvectors, independent of α and β, whereas the eigenvalues μ_i (using Σ_n) are related to λ_i (using Σ_Δ) as follows

$$\mu_i = \frac{\frac{\lambda_i}{2} - (1-\alpha)}{\alpha - \beta}$$

- Since $0 \leq \mu_i$ and $\beta \leq \alpha$ it follows that

$$\beta \leq 1 - \lambda_i/2 \leq \alpha, \quad \forall i$$

  We obtain a lower bound for β by taking the largest value of λ_i, whereas if we us the smallest λ_i we get an upper bound for α.

- Using $\alpha = 1$ and $\beta = 0$ the temporal covariance of Φ

$$\mathrm{Cov}[\Phi(t).\Phi(t+\Delta)] = \mathrm{Cov}[X(t)\Psi, X(t+\Delta)\Psi] \tag{C.8}$$

$$= \Psi^T(\Gamma_s(\Delta) + \Gamma_s(-\Delta))\Psi \tag{C.9}$$

using C.4 and considering only signal correlation with $\Gamma_\delta = 1/2(2\Sigma - \Sigma_\Delta)$

$$\mathrm{Cov}[\Phi(t), \Phi(t+\Delta)] = \Psi^T\Sigma\Psi - \frac{1}{2}\Psi^T\Sigma_\Delta\Psi \tag{C.10}$$

$$= I - \Lambda/2. \tag{C.11}$$

This is an estimation of the temporal covariance of $\phi(t)$ in terms of the generalized eigenvalues λ_i, contained in the diagonal matrix Λ.