Non-linear stochastic methods for discharge prediction

Rafał Wójcik

RAPPORT 88

Sectie Waterhuishouding
Nieuwe Kanaal 11, 6709 PA Wageningen

ISSN 0926-230X

TO MARIËT

AND

TO THE MEMORY OF MY GRANDMOTHER
HANNA CZARNOTA-BOJARSKA

WITH LOVE

This PhD dissertation was sponsored by TEMPUS STEEM Joint European Project nr 11463

The reasearch was carried out at the Department of Environmental Sciences, Sub-Department Water Resources of the Wageningen Agricultural University The Netherlands

Acknowledgments

First and foremost, my deepest gratefulness is due to Paul Torfs for sharing his great ideas with me, for keeping me going with his never ending enthusiasm, for constructive criticism and for his friendship that have been invaluable for me. It is beyond words to express my thankfulness for all the help he has given to me.

I am grateful to Mrs. Ir. E. Wietsma and Prof. S. Ignar for their help in arranging financial assistance for this research from TEMPUS STEEM project.

Also I thank my Parents, my friends and all the people who helped and supported me during the last three years.

Non-linear stochastic models for discharge prediction

by

Rafał Wójcik

M.Sc.eng., Warsaw Agricultural University (1996)

Submitted to the Department of Hydrology and Water Management, Land Reclamation and Environmental Engineering Faculty
for the degree of

Doctor of Philosophy

at the

WARSAW AGRICULTURAL UNIVERSITY

June 1999

Signature of Author ....
Department of Hydrology and Water Management, Land Reclamation and
Environmental Engineering Faculty
18 June 1998

Certified by ....
dr Paul Torfs
Mathematical consultant, Catchment Hydrology Unit, Dept. of Water Resources,
Wageningen Agricultural University, The Netherlands
Research Head

Certified by ....
dr hab. Stefan Ignar
Associate Professor, Dept. of Hydrology and Water Resources, Warsaw Agricultural
University, Poland
Thesis Supervisor

Accepted by ....

Non-linear stochastic models for discharge prediction

by

Rafał Wójcik

Submitted to the Department of Hydrology and Water Management, Land Reclamation and
Environmental Engineering Faculty
on 18 June 1998,
for the degree of
Doctor of Philosophy

Abstract

One of the many types of non-linear black-box models that found applications in prediction of hydrological phenomena are the *probabilistic neural networks* and the *local polynomial state space models.*

Probabilistic neural networks are based upon the Parzen approximation of probability densities by (Gaussian) kernels. The adventages of probabilistic neural networks are that they learn extremly quickly, give probabilistic interpretation of the results and by this not only produce estimation of the mean but also give insight into the other statistics of the errors.

When in higher dimensions the observations tend to cluster around lower dimensional subspaces, the classical approach fails by not being able to take this into account. The solution proposed here is to use a local version, based upon Gaussian kernels with locally estimated covariances.

The above concept resembles the "local and global embedding dimension" used in classical deterministic time series analysis. The idea of state space reconstruction out of a single observable from deterministic system (Takens theorem) provides also a framework for making predictions by local polynomial models. This powerful technique uses local polynomial maps to describe the evolution of points from one state space neighbourhood to the next neighbourhood.

As an example, results on predicting discharges in a small catchment in The Netherlands will be presented. Inputs are lagged discharges. If the time discretisation scale is rather small, and one uses many lags, the input space becomes high dimensional but the observations by the mutual dependence between the components of the input fill only a lower dimensional subspaces. It will be shown that these new techniques offer better results in this case.

Research Head: dr Paul Torfs
Title: Mathematical consultant, Catchment Hydrology Unit, Dept. of Water Resources, Wageningen Agricultural University, The Netherlands

Thesis Supervisor: dr hab. Stefan Ignar
Title: Associate Professor, Dept. of Hydrology and Water Resources, Warsaw Agricultural University, Poland

Contents

List of Figures

List of Tables

Part I

Introduction

Chapter 1

Discharge forecasting in catchment hydrology

Hydrological prediction is the estimate of future states of hydrological phenomena in real time. It comprises technical activities connected with hydrological and non-hydrological subjects, such as network design, data processing, mathematical modelling, remote sensing techniques, telecommunication, operational use of computer systems, etc. Taking this into account, the subject of hydrological forecasting should not be considered as one particular hydrological technique, but as an economic activity using many technological developments, both hydrological and non-hydrological.

The most frequent type of forecast is the flood forecast. Besides, hydrologists are interested in low flow forecasts, in water quality forecasts and in forecasts of the hydrological effects of man-made changes in river catchments. According to a survey of forecasting systems carried out by Water Management Organization (Nemec [43]) the various areas of applications of flow prediction were:

1. Flood protection (43%)

2. Energy (19%)

3. Navigation (12%)

4. Water supply and sanitation (12%)

5. Irrigation (6 %)

6. Water pollution control (4 %)

7. Ice problems (4 %)

The hydrological variables to be predicted were:

1. Surface water level (42%)

2. Discharge (36%)

3. Volume of runoff (21%)

4. Ice, ground water, water quality (seldom)

with forecasting horizons:

1. up to two days: short term (33.5%)

2. two - ten days: medium term (52.5%)

3. beyond ten days: long term (14%)

Hydrological forecasts are estimated by means of hydrological models. Without any pretension to give a complete review of discharge prediction methods, it can be generally said (Watts [64]) that models fall into one of four broad classes:

- physical models
- conceptual models
- empirical models
- black-box models

Since in this thesis, black- box statistical models will be utilized for predicting discharge in small hydrological catchment, it might be desirable to comment briefly on the class of models to which it belongs in comparison to other classes of models.

1.1 Physical models

Physical models attempt to represent all the hydrological processes in the hydrological system in terms of physical laws of water movement (mass balance and energy balance) given by partial differential equations and measured system characteristics. It is theoretically possible to provide all of the input data required by the model directly from field and laboratory experiments. In practice, however, the development of physically based models is complicated; often such models are used only for a limited physical system or for research purposes (e.g. scenario studies of climate change impact on river flow (Grabs [27]). One of the most complicated physical models developed to date is the SHE (System Hydrologique Europeen) model (Abott et al.[4]). This model incorporates one-dimensional (vertical) unsaturated flow, two - dimensional (horizontal) saturated flow and one dimensional river flow. Other modules deal with snow melt, evapotranspiration and chemical transport. SHE is one of the most complete representations of hydrological cycles ever attempted.

The main drawback with physical modeling is that it often incorporates vast amount of parameters that have to be measured in a particular catchment which is expensive process. Besides, physical models usually require enormous computational effort to calculate forecasts of a certain hydrological phenomenon.

1.2 Conceptual models

Conceptual models have been developed to avoid the task of collecting the large amount of experimental data required to run physical models. Their conceptualization is not based on physical processes but on perceived system behavior. Expert-based knowledge of hydrological cycle is divided in subprocesses that are supposed to be of significant importance in the expected use of the model. Each subprocess is then described by simple empirical model of suitable accuracy. Finally, these submodels are integrated to represent the organization of actual flowpaths. In the conteptual models only mass balance is conserved. Parameters of such models may have physical meaning, but due to spatial homogeneity (lumping) they cannot be measured directly and have to be calibrated. The HYRROM (HYdrological Rainfall-Runoff Model) is a good example of conceptual catchment model (Blackie and Eeeles [10]).

1.3 Empirical models

Empirical models are even more remote from physical knowledge. They are based on combinations of simple mathematical operators which are supposed to reasonably describe rainfall-runoff relationships. These models are often developed intuitively, usually from an investigation of simple data sets. Many early hydrological models were empirical (Dooge [16], Nash [42]) but proved to be important in the development of the science of hydrology. Despite their conceptual naivety, the advantage of empirical models is that data requirements for model identification are small, and there are usually only few parameters involved in calibration. One of the most recent developments in this branch of models is GR3J by Edijatno et al. [17].

1.4 Black-box models

Black - box models rely on relationships between input and output data. Even if the exact relationship is unknown, but acknowledged to exist, the model can be "trained" to learn this relationship requiring no prior knowledge of the catchment characteristics. Neither energy nor mass balance solutions are applied in the black-box models. Parameters of such models are not physically interpretable. Thus, those models should be considered as a generalized regression approach to discharge prediction. Obvious examples of linear black boxes are ARMA (Auto-Regressive Moving Average) models (due to Box and Jenkins [13]), and their relatives like ARIMA or ARMAX models. During last few years of rapidly developing science of hydroinformatics it has been recognized, however, that input-output relations·between hydrological variable are often non-linear. Neural networks as a form of non-linear regression are now gaining a lot of attention and are being successfully applied to solve many hydrological problems (see Babovic and Larsen [8]). All discharge forecasting models studied in this thesis fall into non-linear black-box category.

1.4.1 Hybrid approach

Sometimes especially non-linear black -box models are used in combination with physical, conceptual or empirical models. To improve the quality of the predictions black-boxes are trained to predict the residuals of other types of models. A good overview of this kind of updating schemes utilized for forecasting waterlevels of the Rhine river is given by van den Akker and van de Wiel [7].

Chapter 2

Non-linearity in time series - taxonomy

Because in this thesis the main attention will be put on black–box non-linear discharge prediction methods, it is useful to outline different approaches developed towards understanding and modeling real-life time series. Of special importance is the notion of non-linearity, which will be defined in terms of stochastic, deterministic and practical views on analyzed data. For simplicity, only univariate examples will be presented in what follows, but of course these considerations can easily be generalized to multivariate cases. As a common practice in time series analysis, stationarity is assumed. For the sake of convienience, only discrete-time systems are inspected.

2.1 Stochatic view

A classical stochastic approach to time series analysis is based on assumption that the source (or generator) of the investigated time series can be modeled as a combination of the past (lagged) values of the time series and (lagged values of) random noise. The random noise is regarded as time -independent random series having an arbitrary distribution (for a precise definition see Priestley [48] p.101). It is important to stress that term "stochastic" in the title indicates that the random noise is treated as an *internal part* of the analyzed dynamical process (e.g. Eq.(2.2)).

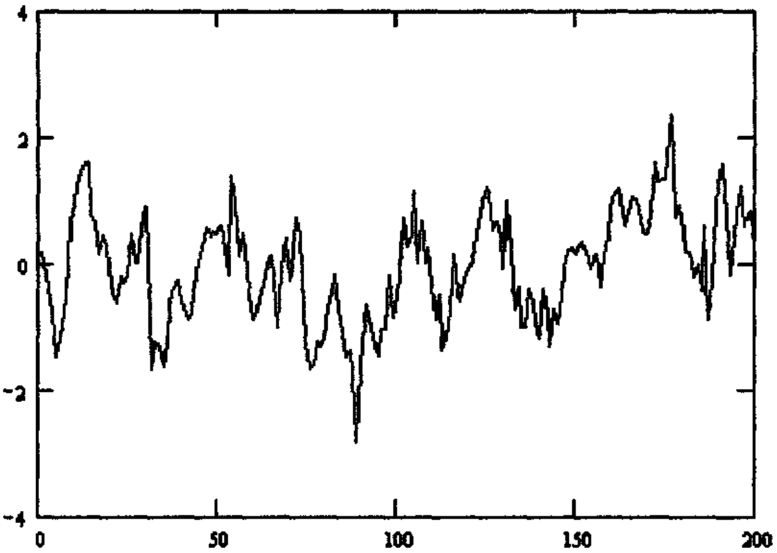

Figure 2-1: A realization of stationary ARMA(1,1) process.

2.1.1 Linear Gaussian stochastic processes

According to Priestley [48] we call stochastic process $\{X_t\}$ linear Gaussian ($\mathcal{LG}$) if it can be represented in one of the following forms:

$$X_t = \sum_{u=0}^{N} \gamma_u \varepsilon_{t-u} \tag{2.1}$$

which is called purely random or MA process, or

$$X_t = \sum_{k=1}^{p} \alpha_k X_{t-k} + \sum_{l=0}^{q} \beta_l \varepsilon_{t-l} \tag{2.2}$$

which is called autoregressive MA, or simply ARMA process (for detailed treatment see Box and Jenkns [13]) of order (p,q); $\gamma_u, \alpha_k, \beta_l$ are constant coefficients and $\{\varepsilon_t\}$ is zero -mean Gaussian noise.

Remark 1 *Any Gaussian process (see e.g. Priestley [48] p.113) can be approximated by Eq.(2.1) or (2.2). Then, Gaussian non-linear processes do not exist.*

2.1.2 Linear non-Gausian processes

There is no consistency in the literature whether processes given by Eq.(2.1) or (2.2) with relaxed Gaussianity assumption on $\{\varepsilon_t\}$ should be referred to as linear or non-linear (Priestley [48], Tong [60]). Hereon, following the convention of Priestley [48], these variations will be denoted as linear non-Gaussian processes or $\mathcal{LNG}$ for short). A typical example is so-called "shot-noise" :

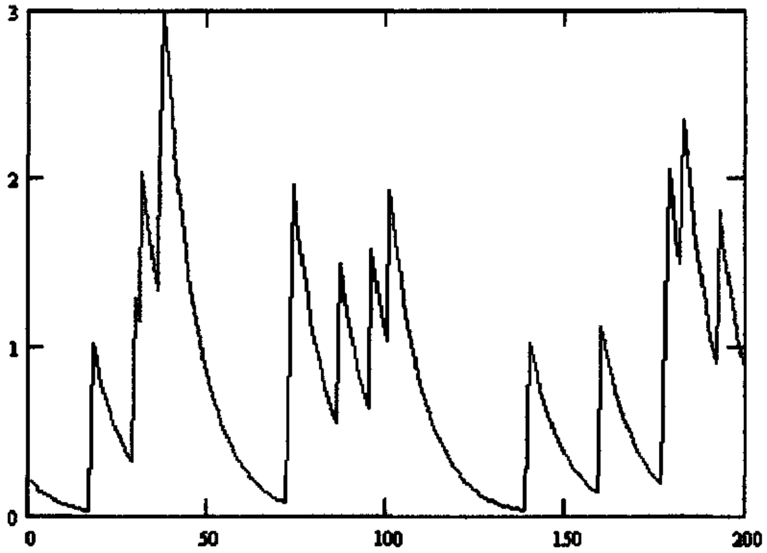

Figure 2-2: A realisation of "shot noise".

$$X_t = 0.9X_{t-1} + e_t \qquad (2.3)$$

where

$$e_t = \begin{cases} 1 \text{ with probability } p \\ 0 \text{ with probability } 1-p \end{cases}$$

Figure 2-2 illustrates this.

2.1.3 Non-linear stochastic processes

In general form non-linear stochastic process can be defined as:

$$X_t = f(X_{t-1}, X_{t-2}, ..., X_{t-N}, e_{t-1}, e_{t-2}, e_{t-M}) \tag{2.4}$$

where e is a noise term and $f(\cdot)$ is a non-linear function. For the sake of example Fig.2-3 shows a realization of bilinear process (Rao et al. [50]) of the form:

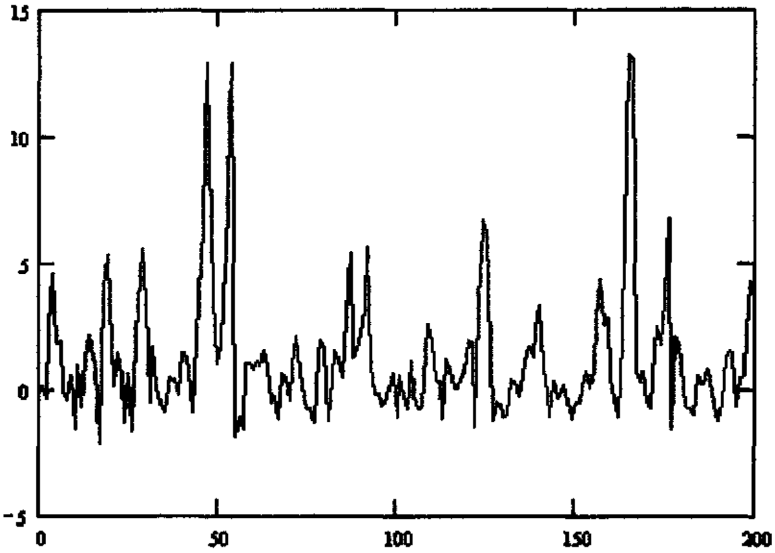

Figure 2-3: A realization of non-linear stochastic process.

$$X_t = 0.4X_{t-1} + 0.6X_{t-1}e_{t-1} + e_t \tag{2.5}$$

with (in this case) Gaussian noise term.

2.2 Deterministic view

The noise term that constitutes the dynamics of stochastic processes can be interpreted in a slightly different way. We assume that the dynamics of the source of our data is purely deterministic, but the observed output series are contaminated by measurement noise. Rephrasing it in terms of deterministic systems theory : we believe that there exists a state variable which uniqely characterizes system's behavior (or uniquely describes input-output relation) at a particular moment in time, however, while measuring output (a function of the state) our measuring device generates some noise. This can be expressed in a set of equations:

$$Z_t = f\left(Z_{t-1}, Z_{t-2}, ..., Z_{t-N}, U_t, U_{t-2}, ..., U_{t-M}\right) \tag{2.6}$$

$$Y_t = h(Z_t) + \varepsilon_t \tag{2.7}$$

where Z_t is the state variable, U_t is an independent known input term, Y_t is the output variable and ε_t is measurement noise, assumed (which is widely accepted convention in the systems theory) to be Gaussian. Formulas (2.6) and (2.7) represent the state equation and the output equation respectively. It should be pointed out that the statistical methods (widely discussed in Section 5.1) associated with this deterministic view to time series aim to reconstruct Z_t directly from Y_t by so-called time delayed embedding.

2.2.1 Linear systems

A system is linear if functions $f(\cdot)$ and $h(\cdot)$ are additive[1] and multiplicative[2]. It is important to signalize that if one reconstructs state space of a linear system from output variable by means of Takens theorem (see Section 5.1.2), the system's dynamics is strictly determined by simple $\mathcal{LG}$ process.An example of the output from periodically driven linear system :

$$Z_t = 0.1Z_{t-1} - 6.8\sin(t-1) \tag{2.8}$$

$$Y_t = Z_t + \varepsilon_t \tag{2.9}$$

is plotted in Fig.2-4

[1] Additivity means : $g(a+b) = g(a) + g(b)$ for any real numbers a and b

[2] Multiplicativity means : $g(ca) = cg(a)$ for any real numbers c and a

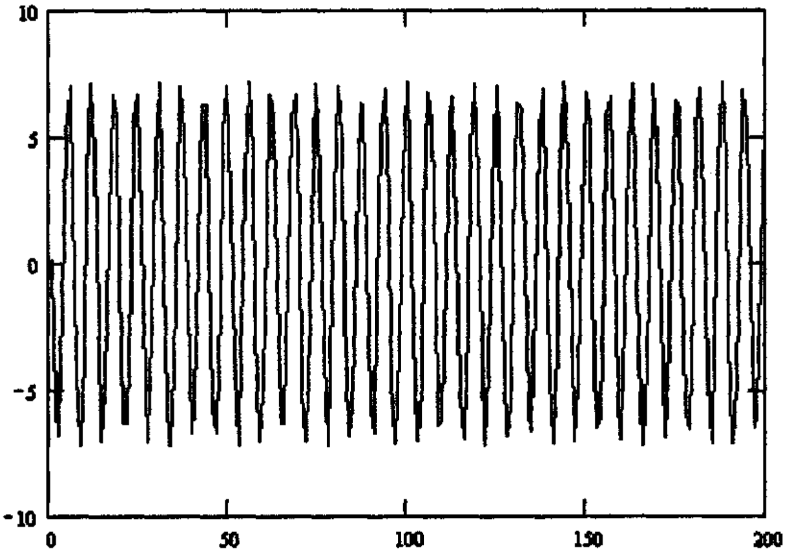

Figure 2-4: Output from a simple linear system.

2.2.2 Non-linear systems

A system is non-linear in weak sense if $h(\cdot)$ is nonlinear function but $f(\cdot)$ is linear. This property will be referred to as $\mathcal{LG}$ *nonlinearity*, since underlying dynamics is linear but state of the system is not observed directly : it undergoes static non-linear transformation. Fig.2-5 shows this effect for output from the system (2.8) when :

$$Y_t = \cos((Z_t + \varepsilon_t)^3) \tag{2.10}$$

Strict (or strong) non-linearity requires that $f(\cdot)$ must be nonlinear. This behavior is recognized as *non-$\mathcal{LG}$ nonlinearity* . A classical example is chaotic[3] Ulam map :

$$Z_t = 4Z_t(1 - Z_t) \tag{2.11}$$

which is set forth in Fig. 2-6

[3] Note that dynamic nonlinearity does not neccecerily imply chaos

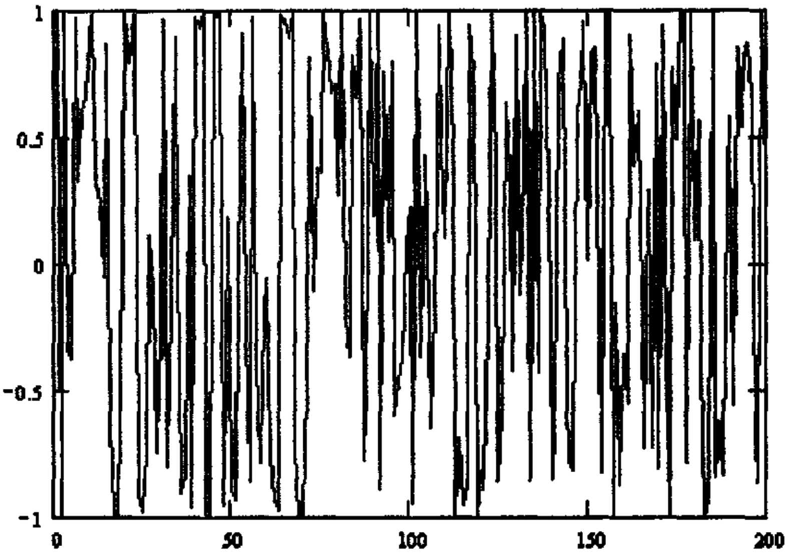

Figure 2-5: Output from a simple linear system after static non-linear transformation.

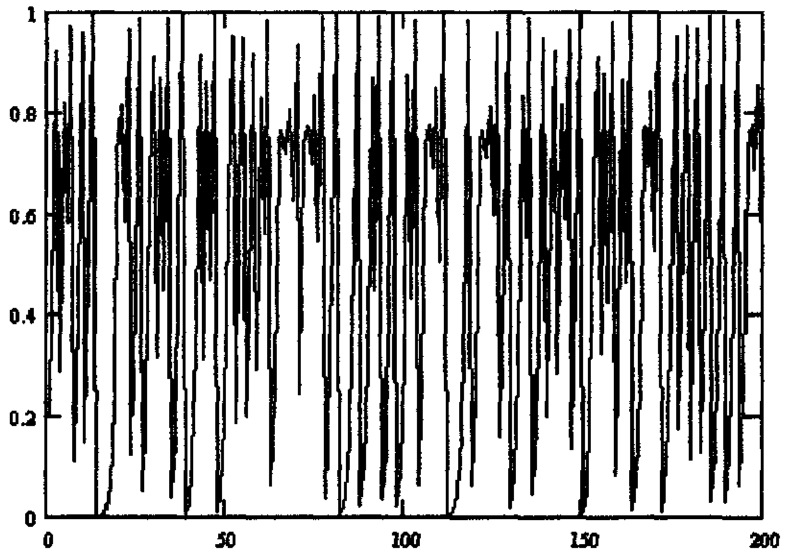

Figure 2-6: Dynamic nonlinear series from Ulam map.

2.3 Practical view on non-linearity in this thesis

From the two above Sections it is clear that non-linearity can appear in many different flavors depending on a point of view to data. A problem of a crucial importance discussed recently in non-linear time series literature (see e.g. Theiler et al.[59], Shreiber [56, 57], Palus [44]) is how to quantitatively detect those different forms of non-linear information. This difficult issue arises especially in a context of building forecasting models of the time series. In other words we ask whether it is worth to fit a non-linear model to data while perhaps simple linear model can be sufficient to make good predictions. In Section 6.3.1 a method for qualitative description of different types (that is $\mathcal{LG}$ or non-$\mathcal{LG}$) of non-linearity in hydrological time series, will be proposed. However, this method gives only some insight into the underlying dynamics and provides no definite answers. This is due to the fact that e.g. some common assumptions imposed on data, like stationarity, may not be fulfilled . One could argue that there are many preprocessing methods to deal with these problems. On the other hand filters like e.g. first order differencing may accidentally destroy some important (for making predictions) aspects of data (Masters [40]). Moreover, no strict algorithms are available to determine which of these preprocessors will serve the best for a particular time series case. Accordingly, a wide margin of subjective (in no sense optimal) choices is left for the analyst.

Therefore in this thesis a practical way of quantification of non-linear information is postulated. We will consider time series as it is without doing any preprocessing, try to fit a simple $\mathcal{LG}$ model and some flexible[4] non- linear models, and eventually compare these techniques in terms of predictability measures. A model that gives the best predictions is optimal. Any improvements in making predictions by non-linear models over $\mathcal{LG}$ models signify the existence of non-linear information contained in time series .

[4] A flexible non-linear model can in principle resemble behavior of linear one provided that linearity assumption holds. Flexibility also implies that the model itself does not require any assumptions on data .

Chapter 3

Objectives

The explicit aims of this investigation were to :

- develope the-state-of-the-art non-linear stochastic prediction model called *local probabilistic neural network* and apply it to discharge prediction
- introduce the *state space analysis methods* (with emphasis to local polynomial models) to catchment hydrology and apply them to discharge prediction
- compare the above two local techniques with each other and with global linear and non-linear forecasting schemes in terms of quality of the predictions

All of the methods studied in this thesis were regarded as autoregression techniques, which implies that discharge forecasts were based only on the past discharges (rainfall intensity was not included as an input variable). Within deterministic framework, this approach is justified by Takens theorem about the state space reconstruction out of a single observable from deterministic system. Besides, this methodology is highly practical since quite often the rainfall information is just missing or incomplete. Additionally, with the help of global and local embedding dimension analysis one is able to determine number of inputs to global and local non-linear models respectively.

As an example of application, results of short term prediction of discharges in a small catchment Hupselse - Beek in The Netherlands will be presented.

3.1 Overview of the dissertation

The thesis is organized as follows. Chapter 4 outlines the central mathematical ideas of Parzen density estimators that will be used to construct local probabilistic neural networks. Furthermore, some aspects of this technique like data-resampling and calculation of the expectation of Mutual Information are shown. Chapter 5 reviews non-linear time series models used in this investigation. Extended treatment is given to state-space reconstruction and prediction methods. Chapter 6 presents case study results of application of all the non-linear techniques in question to the problem of short range discharge prediction. Non-linear aspects of the studied discharge time series are presented in context of predictability characteristics. The two final Chapters address conclusions from the research and future research directions.

3.2 Remarks

Due to large variety and novelty of methods studied, in this thesis there is no global literature review. However, a bibliography is included, which is intended to provide a useful pointers to the literature rather than a complete record of the historical development, especially in the context of Parzen density estimators and neural networks (Sections 4.1.1 and 5.2.2 respectively).

Part II

Methodology

Chapter 4

Novell general statistical tools

In this chapter several new statistical tools , used in this work will be described. The word "general" stands for "no assumptions required" and provides an assumption-free, flexible stochastic framework for time series forecasting. The fundamental building-block for all the methods in question is Parzen probability density estimator (for a classical reference see Wand and Jones [63]). The mathematical material of this Chapter is rather technical but unavoidable for complete understanding of the power of the algorithms. However, the author's intent was to make these concepts clear for practitioners with different backgrounds, so some intuitive framework together with pointers to hydrological literature will be presented as well. Let us start.

4.1 Parzen density estimators

4.1.1 Parzen densities in hydrology

Recently Parzen estimators have been successfully applied in hydrological research. They provide a useful, assumption-free alternative for the parametric approach to probability density function (p.d.f.) estimation. The p.d.f.'s are typically used in hydrology for determination of exceedence probability (e.g. design floods), for regression problems like estimation of water-level - discharge relationships or regression-based forecasting of hydrological time series, and finally for simulation of various hydrological variables like, e.g. rainfall intensity. For example Adamowski [5, 6], performed flood frequency analysis based on kernel methods. In the same spirit Guo Sheng Lian [28], shown that Parzen estimators serve as an alternative way for flood

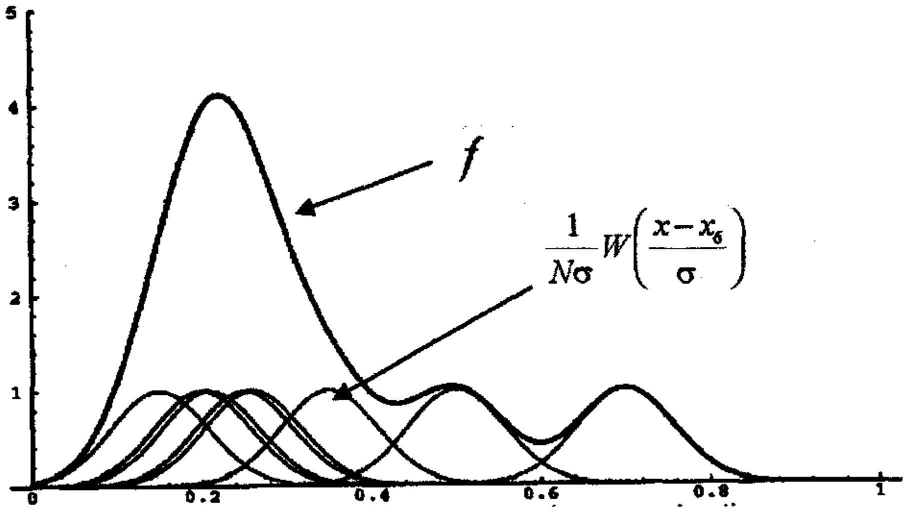

Figure 4-1: An example of kernel p.d.f.

quantile estimation when historical data are available. A gentle overview of potential applications of non-parametric methods together with examples is given by Feluch [19]. More technical issue of bandwidth selection as again related to flood frequency analysis was raised by Lall et al. [35]. Similarly, Rajagopalan et al. [49] studied various bandwidth estimation methods and a possible utilization of boundary kernels with precipitation data. The problem of bandwidth selection was also considered by Sharma et al. [52], who gives a broad comparison of available methods for global and local bandwidths applied to small samples with Gaussian density.

None of the conducted hydrological research, however, addresses the problem of *localization* of non-parametric kernels, which may be useful for building forecasting models of discharges or other hydrological variables. Moreover, as far as non -parametric regression is concerned, none of the reported studies puts emphasis on natural measure of uncertainty of the predictions that emerges naturally from Parzen estimators (see Eq. (4.54) and (4.55)). Consequently, the presentation of this algorithm given in Section 4.1.3 and 4.2 is new to hydrologic literature and provides a novell non-linear basis for hydrological time series prediction.What is more, in Section 5.2.2 I show the parallel implementation of the kernel regression, which is known in the literature as a Probabilistic Neural Network.

Apart from generalized regression problems, the Parzen density estimator can be utilized to estimate Average Mutual Information Function, that can be regarded as a replacement for traditional linear autocorrelation function. Section 4.3 provides a reader with an introduction to this additional analysis tool .

4.1.2 Classical 1-D formulation - an intuitive approach

To estimate a univariate p.d.f. from a random sample $x_1, x_2, ..., x_N$ Parzen's method (Parzen [45]) uses kernel functions $W(d)$, that reach their maximum at $d = 0$ and decrease rapidly when $d > 0$. One of these kernel functions is centered at each sample point x_i, with, the value of each sample's function at a given x being determined by the distance d between x and this sample point. Parzen's p.d.f. estimator f is a scaled sum of these functions for all sample cases (Masters [39]),which can be written as:

$$f(x) = \frac{1}{N\sigma}\sum_{i=1}^{N} W\left(\frac{x - x_i}{\sigma}\right) \tag{4.1}$$

where N is a number of samples in data collection and σ is a scaling parameter. The above formula for 1-D Parzen density is illustrated in Fig.4-1 . The scaling parameter σ (also called a bandwidth or smoothing parameter), specifies a spread of W(d), that is centralized in each x_i. If the value of this parameter is too low, all points x_i, exert too much individual influence on estimated p.d.f. Values which are too high can oversmooth the density estimate, causing a possible loss of details of the density. These effects are best shown in Fig.4-2, which examines the sensitivity of f on σ choice. The issue of automatic selection of σ parameter will be discussed in Section 4.2.4.

There is considerable freedom in choosing $W(d)$ functions. It implies that there are some technical restrictions on their properties (see Parzen [45] or Specht [54]). The two most widely accepted forms of kernels are :

- Gaussian kernel (see upper part of Fig.4-3)

$$W(d) = \frac{1}{\sqrt{2\pi}\sigma} e^{\left(\frac{-d^2}{2\sigma^2}\right)} \tag{4.2}$$

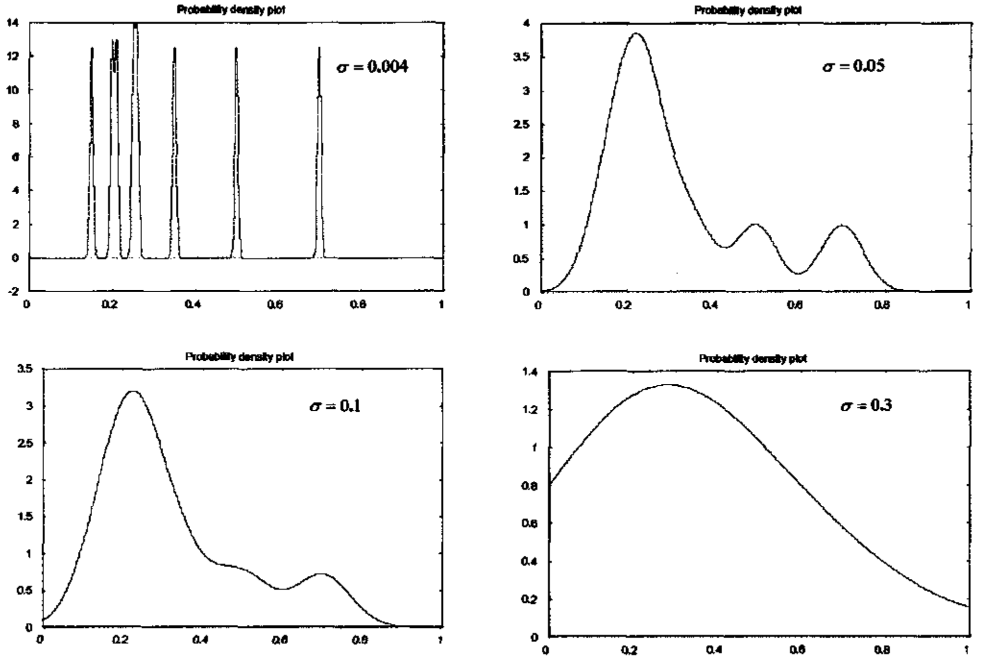

Figure 4-2: Sensitivity of Parzen's p.d.f. estimator to σ choice.

- reciprocal kernel (see lower part of Fig.4-3)

$$W(d) = \frac{1}{\pi\sigma\left[1 + \left(\frac{d}{\sigma}\right)^2\right]} \tag{4.3}$$

Throughout this thesis the Gaussian kernel will be used, however, since it has some nice properties that allow to make certain simplifications which will further be shown. Additionally, it is possible to reformulate the Parzen probability density functions in terms of classical Gaussian kernels. This mathematically gentle connection is the subject of the next Section. Once again, it should be stressed that the original intuition that lies behind the Parzen estimators is shown in Fig.4-1

4.1.3 Extension into higher dimensional spaces

To generalize the Parzen density concept into higher dimensional data spaces , and to introduce the role of covariance matrix in the density estimation, the Parzen densities will be re-derived

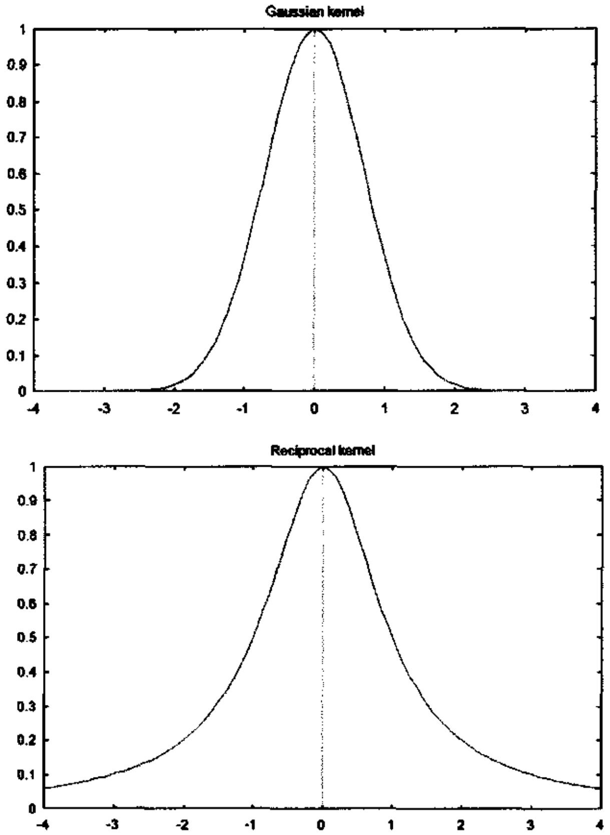

Figure 4-3: Common forms of kernel functions.

on the basis of the classical Gaussian kernels. One should remember however, that *the Parzen densities have nothing to do with Gaussianity assumption on data.* Let us define stochastic variable:

$$\mathbf{X} = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_N \end{bmatrix} \tag{4.4}$$

A typical realization of **X** is denoted by lowercase **x**.

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} \tag{4.5}$$

Than, the multivariate, N- dimensional Gaussian density function is given by:

$$f(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^N |\mathbf{C_{X,X}}|}} \exp\left[-\frac{(\mathbf{x}-\mathbf{m_x})^\top \mathbf{C_{X,X}}^{-1}(\mathbf{x}-\mathbf{m_x})}{2}\right] \tag{4.6}$$

where :

$$\mathbf{m_X} = \int \mathbf{x}\, f(\mathbf{x})\, d\mathbf{x} = \mathrm{E}[\mathbf{X}] = \begin{bmatrix} \mathrm{E}[X_1] \\ \mathrm{E}[X_2] \\ \vdots \\ \mathrm{E}[X_N] \end{bmatrix} \tag{4.7}$$

is an expectation and

$$\begin{aligned} \mathbf{C_{X,X}} &= \int (\mathbf{x}-\mathrm{E}[\mathbf{X}])(\mathbf{x}-\mathrm{E}[\mathbf{X}])^\top f(\mathbf{x})\, d\mathbf{x} && (4.8)\\ &= \mathrm{COV}(\mathbf{X}) && (4.9)\\ &= \mathrm{E}[(\mathbf{X}-\mathbf{m_X})(\mathbf{X}-\mathbf{m_X})^\top] && (4.10)\\ &= \mathrm{E}[\mathbf{X}\mathbf{X}^\top] - \mathbf{m_X}\mathbf{m_X}^\top && (4.11) \end{aligned}$$

is the variance - covariance matrix with the (i,j)-th compontent given by :

$$\begin{aligned} [\mathbf{C_{X,X}}]_{(i,j)} &= \mathrm{E}[(X_i - m_{X_i})(X_j - m_{X_j})] && (4.12)\\ &= \mathrm{COV}[X_i, X_j] && (4.13) \end{aligned}$$

The square root of the quantity :

$$\|\mathbf{x}-\mathbf{m_X}\|^2_{\mathbf{C_{XX}}} = (\mathbf{x}-\mathbf{m_x})^\top \mathbf{C_{X,X}}^{-1}(\mathbf{x}-\mathbf{m_x}) \tag{4.14}$$

is sometimes called *Mahalanobis* distance. Using this notation the density can be rewritten in a mode condensed form :

$$f(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^N\,|\mathbf{C}_{\mathbf{x},\mathbf{x}}|}} \; \exp\left[-\frac{\|\mathbf{x}-\mathbf{m}_{\mathbf{x}}\|^2_{\mathbf{C}_{\mathbf{xx}}}}{2}\right] \tag{4.15}$$

Parzen distributions revisited

We call $f(\mathbf{x})$ a Parzen density if it is the average of K gaussian densities :

$$\begin{aligned} f(\mathbf{x}) &= \frac{1}{K}\sum_i f^{(i)}(\mathbf{x}) && (4.16)\\ &= \frac{1}{K\sqrt{(2\pi)^N}}\sum_i \frac{1}{\sqrt{|\mathbf{C}^{(i)}|}} \; \exp\left[-\frac{\|\mathbf{x}-\mathbf{m}^{(i)}\|^2_{\mathbf{C}^{(i)}}}{2}\right] && (4.17)\end{aligned}$$

where $\mathbf{m}^{(i)}$ are given (local) mean vectors and $\mathbf{C}^{(i)}$ are given covariance matrices and $i = 1, ..., K$ is the density index. The first moment is given by :

$$\begin{aligned} \int \mathbf{x}\, f(\mathbf{x})\, d\mathbf{x} &= \frac{1}{K}\sum_i \int \mathbf{x}\, f^{(i)}(\mathbf{x})\, d\mathbf{x} && (4.18)\\ &= \frac{1}{K}\sum_i \mathbf{m}^{(i)} = \mathbf{m} && (4.19)\end{aligned}$$

The second moment is given by :

$$\int \mathbf{x}\mathbf{x}^\top f(\mathbf{x})d\mathbf{x} = \frac{1}{K}\sum_i \left(\mathbf{C}^{(i)} + \mathbf{m}^{(i)}\mathbf{m}^{(i)\top}\right) \tag{4.20}$$

and the covariance by :

$$\begin{aligned} \mathbf{C} &= \int (\mathbf{x}-\mathbf{m})(\mathbf{x}-\mathbf{m})^\top f(\mathbf{x})\, d\mathbf{x} && (4.21)\\ &= \int \mathbf{x}\mathbf{x}^\top f(\mathbf{x})d\mathbf{x} - \mathbf{m}\mathbf{m}^\top && (4.22)\\ &= \frac{1}{K}\sum_i \left(\mathbf{C}^{(i)}\right) + \left(\frac{1}{K}\sum_i \mathbf{m}^{(i)}\mathbf{m}^{(i)\top}\right) - \left(\frac{1}{K}\sum_i \mathbf{m}^{(i)}\right)\left(\frac{1}{K}\sum_i \mathbf{m}^{(i)}\right)^\top && (4.23)\end{aligned}$$

To put it in words : the covariance is the mean covariance + covariance of the mean

4.1.4 Estimation of Parzen density based on data

The random sample of $N-$ dimensional stochastic variable, consisting of K cases is given by:

$$\mathbf{x}_i = \begin{bmatrix} x_{i,1} \\ x_{i,2} \\ \vdots \\ x_{i,N} \end{bmatrix} \quad \text{for} \quad i = 1, ..., K \tag{4.24}$$

To find a Parzen density such that the data points can be considered as independent realizations following the corresponding probability distribution we have to make choice of Parzen centers $\mathbf{m}^{(i)}$. There are four options available:

1. place $\mathbf{m}^{(i)}$ on discrete grid : this choice, that corresponds to classical bin-based histogramming, is not very effective computationally, since we have to discretize the whole data space
2. draw $\mathbf{m}^{(i)}$ randomly: this is the choice for a technique called *radial basis functions* (Powell [46]). Two strategies are available :
   (a) uniformly
   (b) on the support of the data (regions in space where the probability to see the point is non-zero)
3. draw $\mathbf{m}^{(i)}$ uniformly and iteratively adapt them to the most interesting data regions: this is the choice for *cluster weighted modelling* (Gershenfeld et al. [24])
4. use the observed data points as centers: $\mathbf{m}^{(i)} = \mathbf{x}_i$

We will consider only the fourth choice. This means that the number of components of the Parzen density equals the number of data points K. It yields the following form of the Parzen density estimator:

$$f(\mathbf{x}) = \frac{1}{K\sqrt{(2\pi)^N}} \sum_i \frac{1}{\sqrt{|\mathbf{C}^{(i)}|}} \exp\left[-\frac{\|\mathbf{x}-\mathbf{x}^{(i)}\|^2_{\mathbf{C}^{(i)}}}{2}\right] \tag{4.25}$$

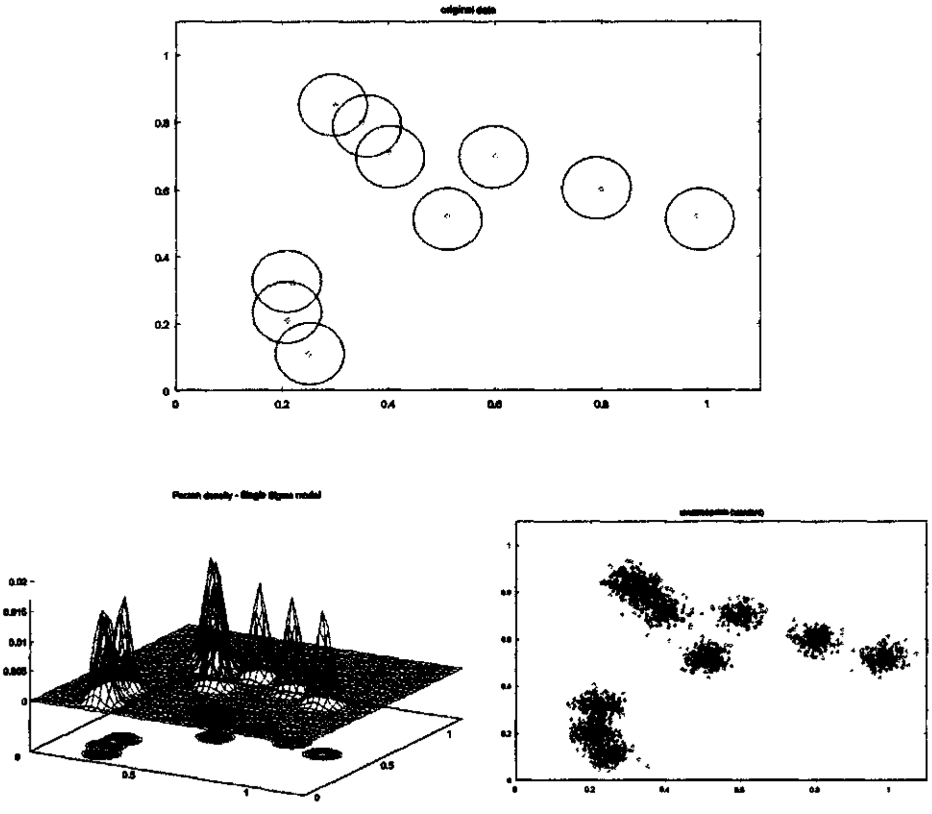

Figure 4-4: An example of standard Parzen density estimator.

4.1.5 The role of the bandwidth and the covariance matrices

The crucial aspect of the Eq.(4.25) is the choice of the covariance matrix:

$$\mathbf{C}^{(i)} = \sigma \cdot \mathbf{CF}^{(i)} \tag{4.26}$$

where $\mathbf{CF}^{(i)}$ are the covariance matrices called covariance frames and σ, as mentined earlier, is a bandwidth. They both play a different role:

- the covariance frames are constructed according to a fixed principle, using data points if needed

- the bandwidth is choosen in such a way that it optimizes a given criterium

Combining the principles above produces the final form of Parzen estimator:

$$
\begin{aligned}
f(\mathbf{x}) &= \frac{1}{K}\sum_i f^{(i)}(\mathbf{x}) && (4.27)\\
&= \frac{1}{K\sqrt{(2\pi)^N}}\sum_i \frac{1}{\sqrt{|\mathbf{C^{(i)}}|}}\ \exp\left[-\frac{\|\mathbf{x}-\mathbf{m^{(i)}}\|^2_{\mathbf{C^{(i)}}}}{2}\right] && (4.28)\\
&= \frac{1}{K\sqrt{(2\pi)^N}}\sum_i \frac{1}{\sqrt{|\sigma\cdot\mathbf{CF^{(i)}}|}}\ \exp\left[-\frac{\|\mathbf{x}-\mathbf{x_i}\|^2_{(\sigma*\mathbf{CF^{(i)}})}}{2}\right] && (4.29)\\
&= \frac{1}{K\sqrt{(2\pi\sigma)^N}}\sum_i \frac{1}{\sqrt{|\mathbf{CF^{(i)}}|}}\ \exp\left[-\frac{\|\mathbf{x}-\mathbf{x_i}\|^2_{\mathbf{C^{(i)}}}}{2\,\sigma}\right] && (4.30)
\end{aligned}
$$

The role of the bandwidth σ in the extreme cases reads :

1. for $\sigma \rightarrow 0$ the probability defined by $f(\mathbf{x};\sigma)$ degenerates to a sum of dirac measures in the observation points (each with mass $1/k$).

2. for $\sigma \rightarrow \infty$ the probability defined by $f(\mathbf{x};\sigma)$ degenerates to an uniform distribution over the whole space.

One considers the following choices of the covariance frames:

- **standard or Single Sigma (SS)**

$$
\mathbf{CF^{(i)}} = \mathbf{I} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} \qquad (4.31)
$$

where **I** is an identity matrix. This choice is global since $\mathbf{CF^{(i)}}$ is independent of i. In this case there is no additional variance-covariance information to the Parzen estimator. When the data is 1-D Eq. (??) corresponds exactly to Eq.(4-1) with Gaussian kernel function. 2-D example of Parzen p.d.f. estimator is shown in Fig.4-4. The data are plotted in the upper picture. The Parzen p.d.f. estimator centers *ciricular* kernel function in each data point : the main support of each kernel function is presented as a small red circle. The radius of each circle represents the line along which $\|\mathbf{x} - \mathbf{x}_i\| = \sigma$. The estimated p.d.f and the points simulated

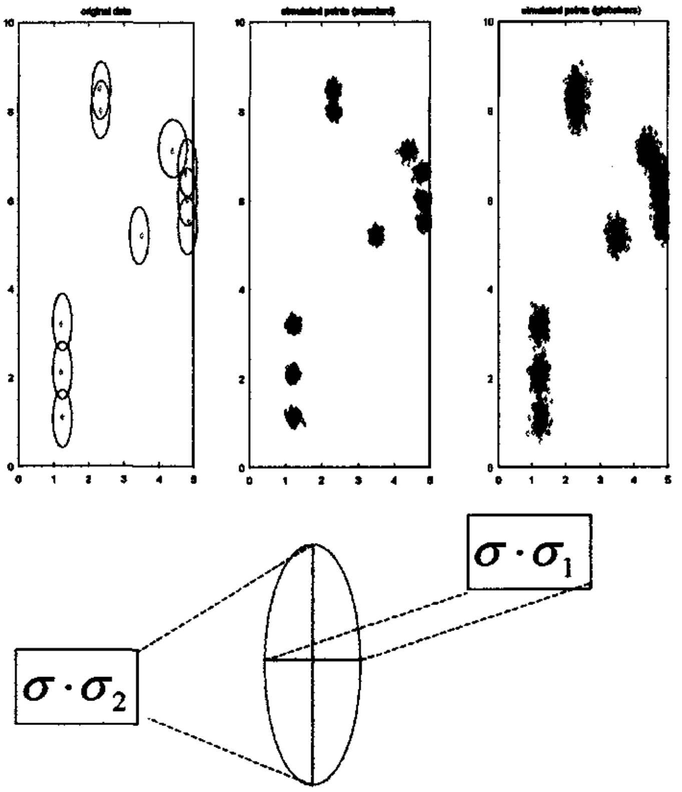

Figure 4-5: An example of global variance Parzen density estimator.

from this density (for details of the simulation procedure see Appendix C) are shown in two lower pictures. The SS model is perfect when the variables are (relatively) independent and have the same variances (or scales) in each direction, i.e. the amount of smoothing is the same in each direction. However, if this property does not hold one has to incorporate the information about the global variance associated with each variable that spanns the data space.

- **global variance or separate variable (SV) :**

$$\mathbf{CF^{(i)}} = \begin{bmatrix} \sigma_1^2 & & & & & \\ & \sigma_2^2 & & & & \\ & & \cdot & & & \\ & & & \cdot & & \\ & & & & \cdot & \\ & & & & & \sigma_N^2 \end{bmatrix} \tag{4.32}$$

where $\sigma_1^2, \sigma_2^2, ..., \sigma_N^2$ are the variances of corresponding $x_1, x_2, ..., x_N$ variables and :

$$\mathbf{CF^{(n)}}_{i,i} = \sigma_i^2 = \frac{1}{K}\sum_k (x_{k,i} - \overline{x_i})^2 \tag{4.33}$$

$$\overline{x_i} = \frac{1}{K}\sum_k x_{k,i} \tag{4.34}$$

where $x_{i.k}$ the k-th component of the i-th data point $\mathbf{x_i}$.

This is again global choice. As an opposite to the SS model, however, the SV is suitable for the situation when the variables do not have the same variances. This time, the p.d.f. estimator centers an elliptical kernel aligned parallel to the coordination system axes. Lengths of the kernel's axes are determined by standard deviations $\sigma_1, \sigma_2, ..., \sigma_N$ of the variables (Eq.(4.55)) multiplied by scaling factor σ. This is set forth in Fig. 4-5 where the SV model of p.d.f is fitted to the data collection shown in the left picture. In the middle picture points drawn from the SS model of the p.d.f are shown. One can see that this model, by not being able to deal with non-uniform variances of the variables, fails to detect the vertical structure in data. Instead, it introduces some artifacts in the p.d.f. : the circular spots representing artificial peaks of the Parzen density. On the other hand, the SV model (right picture) clearly mimics the underlying structure of the p.d.f. by doing different amount of smoothing in each coordinate direction.

- **global covariance (GC):**

$$\mathbf{CF}^{(\mathbf{i})} = \begin{bmatrix} c_{1,1} & c_{1,2} & \cdots & c_{1,N} \\ c_{2,1} & c_{2,2} & \cdots & c_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ c_{N,1} & c_{N,2} & \cdots & c_{N,N} \end{bmatrix} \tag{4.35}$$

where $c_{i,j}$ are the covariances between the variables estimated as:

$$\mathbf{CF}^{(\mathbf{n})}{}_{i,j} = c_{i,j} = \frac{1}{K}\sum_k (x_{k,i} - \overline{x_i})(x_{k,j} - \overline{x_j}) \tag{4.36}$$

$$\overline{x_i} = \frac{1}{K}\sum_k x_{k,i} \tag{4.37}$$

This covariance frame is appropriate in situation when one wishes to smooth the density in directions different to those of coordinate axes (Wand and Jones [63]). In other words one gains an extra degree of freedom by being able to *rotate* an elliptical kernel introduced in SV model. The axes of the ellipse are now aligned along *eigendirections* of the global covariance matrix and their lengths are proportional to square roots of *eigenvalues* (denoted as $d_1, d_2, ..., d_N$) associated with each eigendirection. This situation is shown in Fig. 4-6. In the top picture, points represent values of bivariate random sample. The Parzen estimator is then formed by centering a bivariate kernel function around each point. These kernels, as in the previous examples, are represented by the red elliptical boundary contours. The heights of the kernels are then averaged to form the Parzen density estimator. Then, the approximated distribution is again represented by points drawn from it. This is shown in lower picture of Fig. 4-6. The pattern present in the random sample is clearly reflected.

Wand and Jones [63] or Silverman [53], suggested that the GC model is the most general one. In fact it is not. When (in higher dimensions) observations tend to cluster around lower dimensional subspaces with a local aspect, this classical approach fails by not being able to take this into account. The solution proposed in this thesis is to use *locally* (varying with i) estimated covariances. One considers two local frames:

- **local variance (LV)** :

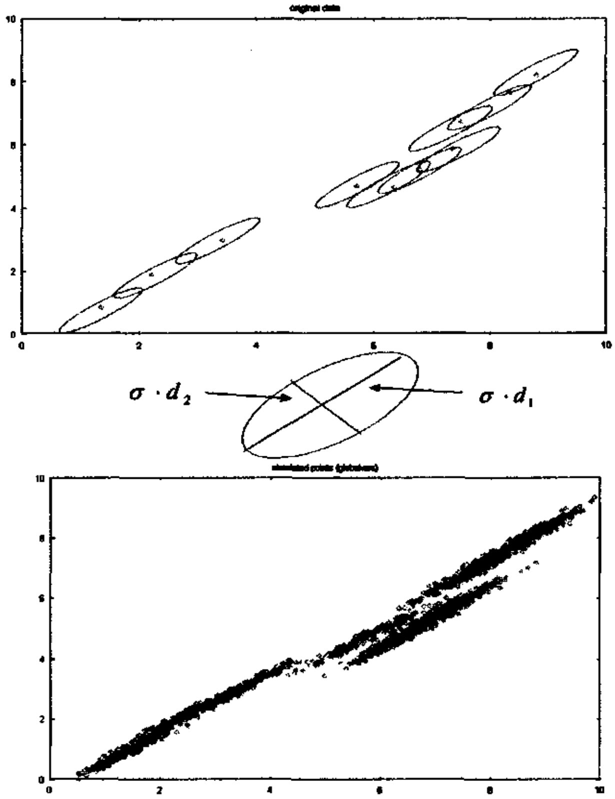

Figure 4-6: Construction and simulation of bivariate kernel (Parzen) density estimate. Upper panel: kernel mass being centered about each observation. Lower panel: points simulated according to the resulting density estimate.

$$\mathbf{C^{(i)}} = \begin{bmatrix} \sigma(i)_1^2 & & & & & \\ & \sigma(i)_2^2 & & & & \\ & & \cdot & & & \\ & & & \cdot & & \\ & & & & \cdot & \\ & & & & & \sigma(i)_N^2 \end{bmatrix} \tag{4.38}$$

where

$$\mathbf{CF^{(n)}}_{i,i} = \sigma^2(n)_i = \frac{\sum_{k \neq n} v_n(\mathbf{x_k})\,(x_{k,i} - x_{n,i})^2}{\sum_{k \neq n} v_n(\mathbf{x_k})} \tag{4.39}$$

and

- **local covariance (LC)** :

$$\mathbf{CF^{(i)}} = \begin{bmatrix} c(i)_{1,1} & c(i)_{1,2} & \cdots & c(i)_{1,N} \\ c(i)_{2,1} & c(i)_{2,2} & \cdots & c(i)_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ c(i)_{N,1} & c(i)_{N,2} & \cdots & c(i)_{N,N} \end{bmatrix} \tag{4.40}$$

In both LV and LC cases the $\mathbf{CF^{(i)}}$ matrix is calculated individually for each sample point $\mathbf{x}_i$ so unlike in Eq.(4.31), (4.32) and (4.35) index i starts to play a role. Locality means that we use, as in the previous case, a kind of neighborhood notion to approximate $\mathbf{CF^{(n)}}$:

$$\mathbf{CF^{(n)}}_{i,j} = c(n)_{i,j} = \frac{\sum_{k\neq n} v_n(\mathbf{x}_k)\,(x_{k,i} - x_{n,i})\,(x_{k,j} - x_{n,j})}{\sum_{k\neq n} v_n(\mathbf{x}_k)} \tag{4.41}$$

In the those two constructions of frames, one needs a weighting function $v_n(\cdot)$. Thre are some possible choices :

1. a global ciricular one:

$$v_n(\mathbf{x}) = \begin{cases} 1 \ \text{if} \ \ \|\mathbf{x} - \mathbf{x}_n\| \leq R \\ 0 \ \text{otherwise} \end{cases} \tag{4.42}$$

   where R is a global fixed radius. The obvious advantage of this choice is small computational effort. In some regions of the data space the situation may happen, however, that $\sum_k v_n(\mathbf{x}) = 0$ which is a potential snag of the method.

2. a nearest-neighbor one:

$$v_n(\mathbf{x}) = \begin{cases} 1 \ \text{if} \ \ \|\mathbf{x} - \mathbf{x}_n\| \leq R_n^{(\varphi)} \\ 0 \ \text{otherwise} \end{cases} \tag{4.43}$$

   where $R_n^{(\varphi)}$ is a local radius within which φ nearest-neighbors of the point $\mathbf{x}$ are included. This approach requires a nearest-neighbor search procedure which is computationally slow and difficult to program.

3. a global exponential one:

$$v_n(\mathbf{x}) = \exp(-\alpha \|\mathbf{x} - \mathbf{x}_n\|^2) \tag{4.44}$$

   This computationally effective solution assures that if point $\mathbf{x}_k$ lies far away from $\mathbf{x}_n$ the weight $v_n(\mathbf{x}_k)$ related to $\mathbf{x}_k$ will be very small so $\mathbf{x}_k$ will not be effectively influencing $\mathbf{CF^{(n)}}$ calculation in $\mathbf{x}_n$. The parameter α determines the decrease rate of weight function $v_n(\cdot)$. The higher the α value[1] the faster the decrease of $v_n(\cdot)$ will be and, as a consequence, the smaller number of neighboring points will be used for $\mathbf{C^{(i)}}$ estimation.

4. a local exponential one. Let β be a (global) fixed number, then let α_n be such that

$$\sum_{n \neq k} \exp\left[-\alpha_n \sum_i (x_{k,i} - x_{n,i})^2\right] = \beta \tag{4.45}$$

   then :

$$v_n(\mathbf{x}) = \exp(-\alpha_n \|\mathbf{x} - \mathbf{x}_n\|^2) \tag{4.46}$$

Remark 2 *If $\beta = K - 1$ then all $\alpha_n \equiv 0$ and accordingly all $v_n(\mathbf{x_k}) \equiv$1. In consequence* $\mathbf{CF^{(n)}}$ *degenerates to the global case. So, β controls the locality of the estimation of each α_n : the smaller the value of this parameter the more local the approximation becomes.*

In this study only the last option was used. Unlike the previous suggestions of parametrization of this matrix (discussed also widely by Sharma et al. [52]) the method presented here ensures that kernels are locally oriented along eigendirections of $\mathbf{CF^{(n)}}$. Hence, apart from the possibility of rotation of each elliptical kernel, the kernel's contours or each ellipse's axis can also locally change their length. This is best illustrated in Fig. 4-7. By the local covariance matrix implementation one is also able to detect local lower dimensional structures (if they exist) in high dimensional data space.

4.1.6 An illustrative example

The consequences of this fact have enormous impact on making predictions of a certain physical phenomena .We will evoke this issue in Section 6.4 where it will be practically applied to

[1] Notice that if $\alpha = 0$ we have a GC model.

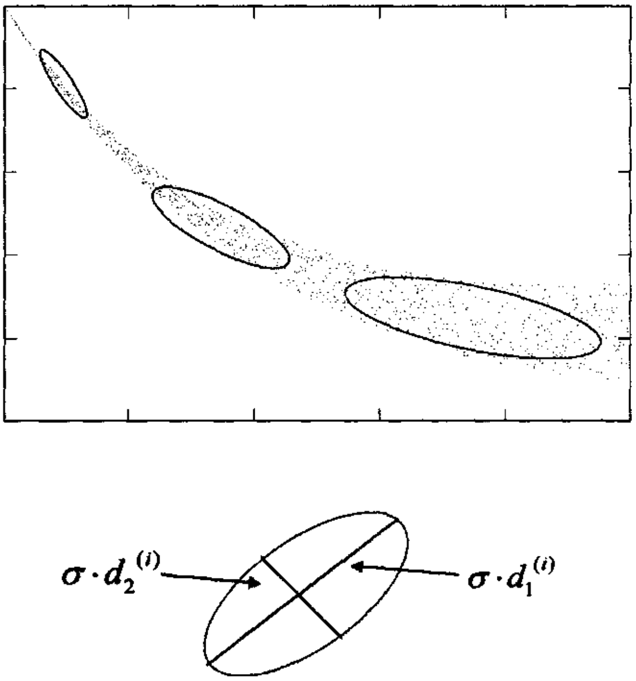

Figure 4-7: An example of 2-D sample space with kernels aligned along eigendirections of local covariance matrix $\mathbf{C}^{(i)}$.

discharge forecasting. Now to build up an intuition again, we will consider an example of a three dimensional data set with locally lower dimensional structure called Lissajous curve. In Fig.4-8 the data points lying along this fancy object are plotted. Figures 4-8 b), c), d), e), f) show points drawn from 3-D Parzen distribution estimated on the basis of the original data (case a). Each of these pictures represents the density estimated using the following choice of the covariance frame SS, SV, GC, LV, LC respectively. One can see that Fig. 4-8 b), c), d) look a bit cloudy, since all local properties of the geometric object are missed. The global approach tends to blur the local structure of the data set by spreading out the probability mass in globally adjusted hyper-ellipses around each observation point. This situation improves dramatically when locally estimated frames (Fig.4-8 e), f)) are used : the Parzen density is more concentrated on the structure present in this data set: Fig. 4-8 f) closely matches Fig.4-8 a).

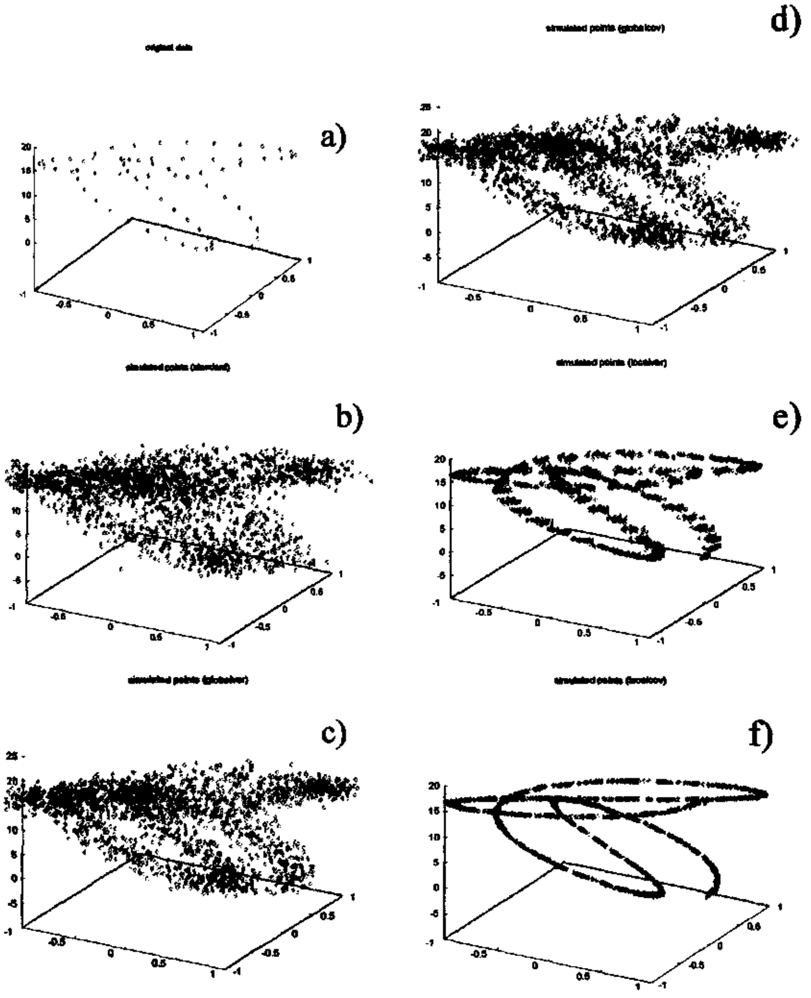

Figure 4-8: Choice of covariance frames is crucial, especially when lower dimensional structures occur in higher dimensional data space. Here, the parametric 1-D Lissajous curve is embedded in 3-D data space.

4.2 Generalized regression

Parzen densities can also be applied to construct generalized regression estimator. For clarity of our considerations, Parzen estimators will now be rephrased in terms of multidimensional input-output mapping framework.

4.2.1 Two component (input - output) Parzen distribution

Two multivariate stochastic variables $\mathbf{X}$ and $\mathbf{Y}$ are said to have a combined Parzen density if their density function can be written as :

$$f_{\mathbf{XY}}(\mathbf{x},\mathbf{y}) = \frac{1}{K}\sum_i f^{(i)}_{\mathbf{XY}}(\mathbf{x},\mathbf{y}) \tag{4.47}$$

where, for each i, $f^{(i)}_{\mathbf{XY}}$ is a Gaussian density function with parameters :

$$\mathbf{m}^{(i)} = \begin{bmatrix} \mathbf{m}^{(i)}_{\mathbf{X}} \\ \mathbf{m}^{(i)}_{\mathbf{Y}} \end{bmatrix} \tag{4.48}$$

$$\mathbf{C}^{(i)} = \begin{bmatrix} \mathbf{C}^{(i)}_{\mathbf{XX}} & \mathbf{C}^{(i)}_{\mathbf{XY}} \\ \mathbf{C}^{(i)}_{\mathbf{YX}} & \mathbf{C}^{(i)}_{\mathbf{YY}} \end{bmatrix} \tag{4.49}$$

If $\mathbf{X}$ and $\mathbf{Y}$ are combined Parzen then each marginal is also Parzen, e.g. :

$$f_{\mathbf{X}}(\mathbf{x}) = \frac{1}{K}\sum_i f^{(i)}_{\mathbf{X}}(\mathbf{x}) \tag{4.50}$$

where $f^{(i)}$ has parameters $\mathbf{m}^{(i)}_{\mathbf{X}}$ and $\mathbf{C}^{(i)}$.

4.2.2 Conditional Parzen distribution

Let $\mathbf{X}$ and $\mathbf{Y}$ be combined Parzen, then the conditional density of $\mathbf{Y}$ given $\mathbf{X}$ can be written as :

$$f_{\mathbf{Y}|\mathbf{X}=\mathbf{x}}(\mathbf{y}) = \frac{\sum_i f^{(i)}_{\mathbf{XY}}(\mathbf{x},\mathbf{y})}{\sum_i f^{(i)}_{\mathbf{X}}(\mathbf{x})} \tag{4.51}$$

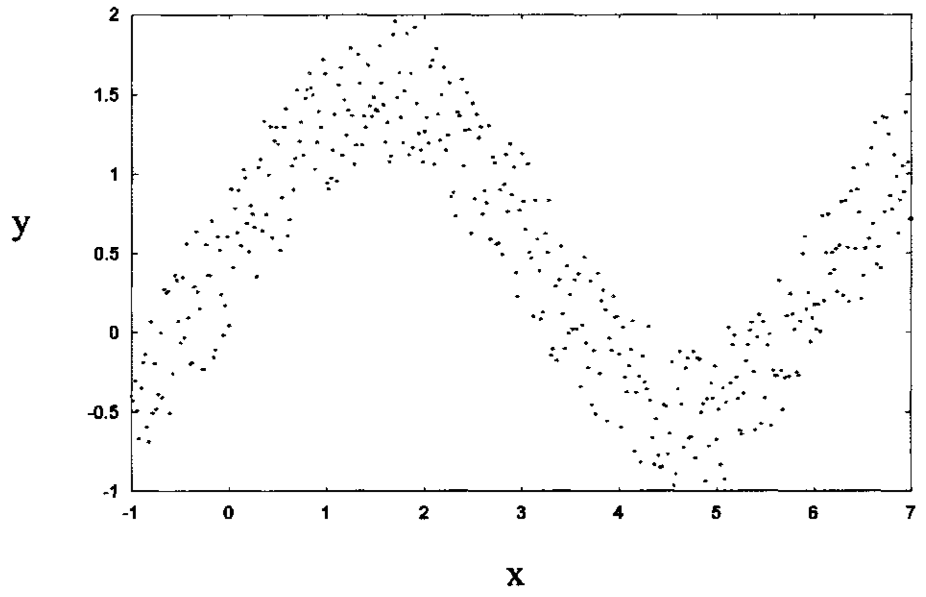

Figure 4-9: Noisy sinusoid.

$$= \frac{\sum_i f_{\mathbf{X}}^{(i)}(\mathbf{x})\, f_{\mathbf{Y}|\mathbf{X}=\mathbf{x}}^{(i)}(\mathbf{y})}{\sum_i f_{\mathbf{X}}^{(i)}(\mathbf{x})} \tag{4.52}$$

The conditional density is thus a weighted (with weighting factors $f_{\mathbf{X}}^{(i)}(\mathbf{x})$ depending on the condition $\mathbf{x}$) of the conditional densities of the components.

4.2.3 Conditional Parzen moments

From elementary statistics it is well known that the best predicted value for $\mathbf{y}$ (in the sense of minimum expected squared error) is its conditional expectation given $\mathbf{x}$. The function $\mathbf{y}(\mathbf{x}) = \mathrm{E}[\mathbf{Y}|\mathbf{X} = \mathbf{x}]$ is called *generalized regression curve* and is given by:

$$\begin{aligned} \mathbf{y}(\mathbf{x}) &= \mathrm{E}[\mathbf{Y}|\mathbf{X}=\mathbf{x}] = \mathbf{m}_{\mathbf{Y}|\mathbf{X}=\mathbf{x}} = \int \mathbf{y}\, f_{\mathbf{Y}|\mathbf{X}=\mathbf{x}}(\mathbf{y})\, d\mathbf{y} \\ &= \frac{\sum_i f_{\mathbf{X}}^{(i)}(\mathbf{x}) \int \mathbf{y}\, f_{\mathbf{Y}|\mathbf{X}=\mathbf{x}}^{(i)}(\mathbf{y})\, d\mathbf{y}}{\sum_i f_{\mathbf{X}}^{(i)}(\mathbf{x})} \\ &= \frac{\sum_i f_{\mathbf{X}}^{(i)}(\mathbf{x})\, \mathbf{m}_{\mathbf{Y}|\mathbf{X}=\mathbf{x}}^{(\mathbf{i})}}{\sum_i f_{\mathbf{X}}^{(i)}(\mathbf{x})} \end{aligned} \tag{4.53}$$

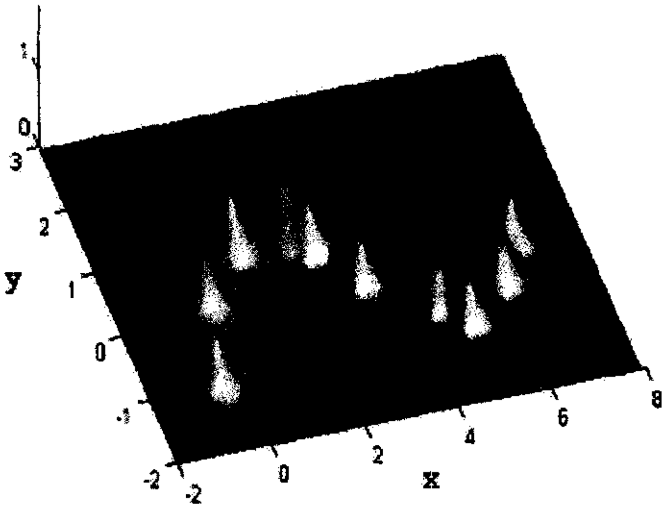

Figure 4-10: Construction of 2-D input-output Parzen density.

To assess the uncertainty of the above predictor we calculate conditional covariance[2] :

$$\begin{aligned}
\mathrm{COV}\left[\mathbf{Y}|\mathbf{X}=\mathbf{x}\right] &= \mathrm{E}[(\mathbf{Y}-\mathbf{m}_{\mathbf{Y}|\mathbf{X}=\mathbf{x}})^2|\mathbf{X}=\mathbf{x}] = \int (\mathbf{y}-\mathbf{m}_{\mathbf{Y}|\mathbf{X}=\mathbf{x}})^2\, f_{\mathbf{Y}|\mathbf{X}=\mathbf{x}}(\mathbf{y})\, d\mathbf{y} \\
&= \frac{\sum_i f_{\mathbf{X}}^{(i)}(\mathbf{x})\, \int(\mathbf{y}-\mathbf{m}_{\mathbf{Y}|\mathbf{X}=\mathbf{x}})^2\, f_{\mathbf{Y}|\mathbf{X}=\mathbf{x}}^{(i)}(\mathbf{y})\, d\mathbf{y}}{\sum_i f_{\mathbf{X}}^{(i)}(\mathbf{x})} \qquad (4.54) \\
&= \frac{\sum_i f_{\mathbf{X}}^{(i)}(\mathbf{x}) \left\{\mathbf{C}_{\mathbf{YY}|\mathbf{X}}^{(\mathrm{i})} + \left(\mathbf{m}_{\mathbf{Y}|\mathbf{X}=\mathbf{x}}^{(\mathrm{i})} - \mathbf{m}_{\mathbf{Y}|\mathbf{X}=\mathbf{x}}\right)\left(\mathbf{m}_{\mathbf{Y}|\mathbf{X}=\mathbf{x}}^{(\mathrm{i})} - \mathbf{m}_{\mathbf{Y}|\mathbf{X}=\mathbf{x}}\right)^{\mathrm{T}}\right\}}{\sum_i f_{\mathbf{X}}^{(i)}(\mathbf{x})}
\end{aligned}$$

and as a measure of this uncertainty (to remain in agreement with original units of $\mathbf{y}$) we simply take ± standard deviation:

$$\mathrm{SD}_i\left[\mathbf{Y}|\mathbf{X}=\mathbf{x}\right] = \pm\sqrt{\mathrm{COV}_{ii}\left[\mathbf{Y}|\mathbf{X}=\mathbf{x}\right]} \qquad (4.55)$$

It is instructive to see how non-parametric regression based on Parzen densities works at solving a problem. Let us use a simple sinusoid (Fig.4-9) with some extra random noise added. The

[2] **For definition of the matrix $\mathbf{C}_{\mathbf{YY}|\mathbf{X}}^{(\mathrm{i})}$ in the Eq. () below see Appendix**

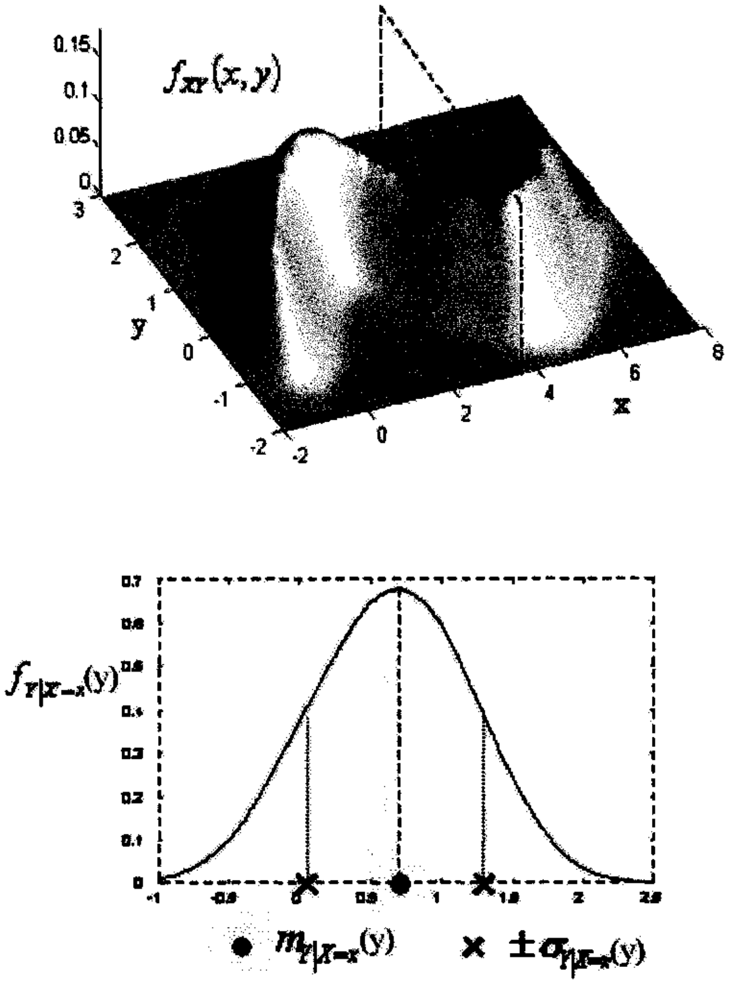

Figure 4-11: Making predictions with Parzen densities.

first step is to estimate an input-output Parzen density. As we know from the beginning of this Chapter we place Gaussian kernel in each of the data points (see Fig. 4-10 - for clarity only a few Gaussians are shown), we add them up and rescale to eventually obtain density approximation shown in the upper panel of Fig. 4-11. Afterwards, in point we want to have an estimate of the dependent variable $\mathbf{y}$, we make a projection of $f_{\mathbf{XY}}(\mathbf{x},\mathbf{y})$ on the plane indicated by dashed line to obtain conditional density[3] $f_{\mathbf{Y}|\mathbf{X}=\mathbf{x}}(\mathbf{y})$ (see lover panel of Fig. 4-11). Our prediction is then simply the mass center of this density and its standard deviation serves

[3] This density is of course rescaled to integrate to unity

as the symmetric uncertainty measure of the forecast. The resulting estimate of the overall regression curve together with the standard deviation bands is shown in Fig. 4-12 .

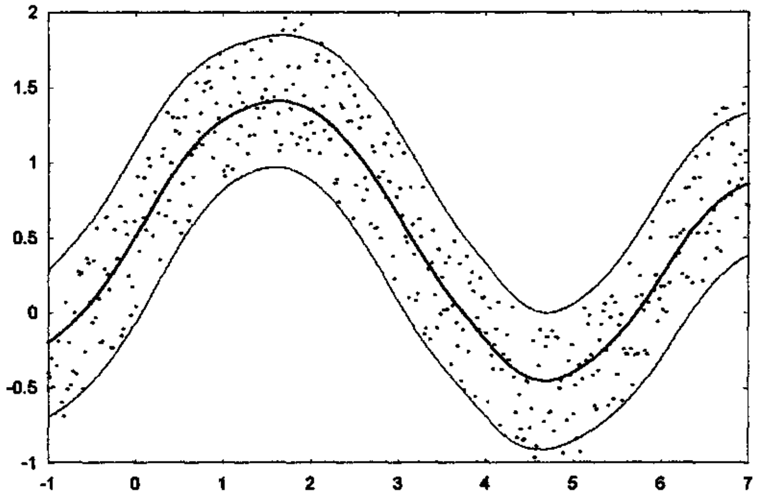

Figure 4-12: Non-parametric regression.

It should be stressed out that the uncertainty corridor (area between black lines) easily adapts to the situation when noise amplitude increases along x axis (see Fig. 4-13).

Looking back at the Eq.(4.53), one notes that in two component case, the bandwidth does not intervene in the formula for the i-th local conditional expectation:

$$\begin{aligned}\mathbf{m}^{(\mathbf{i})}_{\mathbf{Y}|\mathbf{X}=\mathbf{x}} &= \mathbf{y_i} + \mathbf{C}^{(\mathbf{i})}_{\mathbf{YX}}\mathbf{C}^{(\mathbf{i})}_{\mathbf{XX}}{}^{-1}(\mathbf{x}-\mathbf{x_i}) \\ &= \mathbf{y_i} + \left(\sigma^2\mathbf{C}^{(\mathbf{i})}_{\mathbf{YX}}\right)\left(\sigma^2\mathbf{C}^{(\mathbf{i})}_{\mathbf{XX}}\right)^{-1}(\mathbf{x}-\mathbf{x_i}) \\ &= \mathbf{y_i} + (\sigma^2\sigma^{-2})\mathbf{CF}^{(\mathbf{i})}_{\mathbf{YX}}\mathbf{CF}^{(\mathbf{i})}_{\mathbf{XX}}{}^{-1}(\mathbf{x}-\mathbf{x_i}) \\ &= \mathbf{y_i} + \mathbf{CF}^{\mathbf{i}}_{\mathbf{YX}}\mathbf{CF}^{(\mathbf{i})}_{\mathbf{XX}}{}^{-1}(\mathbf{x}-\mathbf{x_i})\end{aligned} \tag{4.56}$$

so that the formula for global conditional mean becomes :

$$\mathbf{m}_{\mathbf{Y}|\mathbf{X}=\mathbf{x}} = \frac{\sum_i f^{(i)}_{\mathbf{X}}(\mathbf{x};\sigma)\ \mathbf{m}^{(\mathbf{i})}_{\mathbf{Y}|\mathbf{X}=\mathbf{x}}}{\sum_i f^{(i)}_{\mathbf{X}}(\mathbf{x};\sigma)}$$

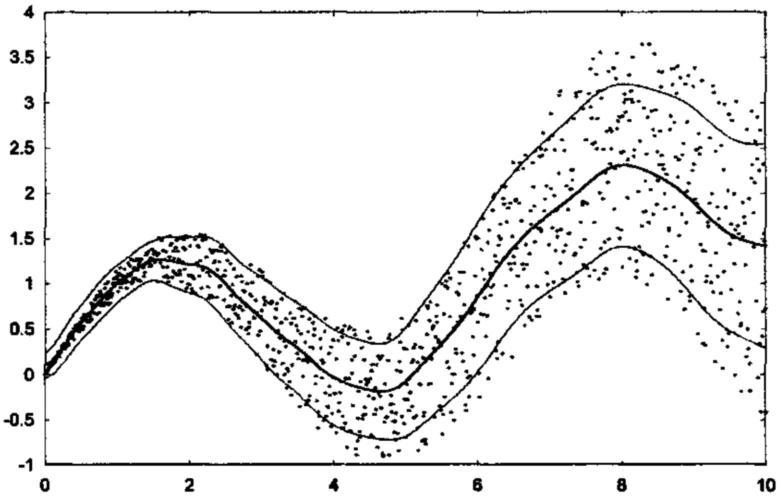

Figure 4-13: Uncertainty bands are expanding together with increase of noise amplitude.

$$\begin{aligned}
&= \frac{\sum_i f_{\mathbf{X}}^{(i)}(\mathbf{x};\sigma)\left[\mathbf{y_i} + \mathbf{CF}^{\mathbf{i}}_{\mathbf{YX}} {\mathbf{CF}^{(\mathbf{i})}_{\mathbf{XX}}}^{-1}(\mathbf{x}-\mathbf{x_i})\right]}{\sum_i f_{\mathbf{X}}^{(i)}(\mathbf{x};\sigma)} \\
&= \frac{\sum_i \frac{1}{\sqrt{|\mathbf{CF}^{(\mathbf{i})}|}} \exp\left[-\frac{\|\mathbf{x}-\mathbf{x_i}\|^2_{\mathbf{CF}^{(\mathbf{i})}}}{2\,\sigma^2}\right]\left[\mathbf{y_i} + \mathbf{CF}^{\mathbf{i}}_{\mathbf{YX}} {\mathbf{CF}^{(\mathbf{i})}_{\mathbf{XX}}}^{-1}(\mathbf{x}-\mathbf{x_i})\right]}{\sum_i \frac{1}{\sqrt{|\mathbf{CF}^{(\mathbf{i})}|}}\left[-\frac{\|\mathbf{x}-\mathbf{x_i}\|^2_{\mathbf{CF}^{(\mathbf{i})}}}{2\,\sigma^2}\right]} \qquad (4.57) \\
&= \frac{\sum_i \mathbf{w}^{(\mathbf{i})}(\mathbf{x},\sigma)\,\hat{\mathbf{y}}^{(i)}(\mathbf{x})}{\sum_i \mathbf{w}^{(\mathbf{i})}(\mathbf{x},\sigma)}
\end{aligned}$$

We can interpret this last formula as follows :

1. for every i we have a *linear* prediction model, that gives the following prediction for the y-value given the x-value :

$$\hat{\mathbf{y}}^{(i)}(\mathbf{x}) = \left[\mathbf{y_i} + \mathbf{CF}^{\mathbf{i}}_{\mathbf{YX}} {\mathbf{CF}^{(\mathbf{i})}_{\mathbf{XX}}}^{-1}(\mathbf{x}-\mathbf{x_i})\right] \qquad (4.58)$$

2. the final prediction is *weighted average* of the predictions made by these local linear models

3. the weighting factors or marginal densities are *local*, i.e. $\mathrm{w}(\mathbf{x};\sigma)$ does depend on $\mathbf{x}$

$$\mathrm{w}^{(i)}(\mathbf{x};\sigma) = \frac{1}{\sqrt{|\mathbf{CF}^{(i)}|}} \exp\left[-\frac{\|\mathbf{x}-\mathbf{x_i}\|^2_{\mathbf{CF}^{(i)}}}{2\,\sigma^2}\right] \tag{4.59}$$

1. the *only* function of the bandwidth σ is to *control this locality*, the smaller the σ, the more local the ultimate prediction, the larger the σ, the more global the ultimate prediction.

This last point is also illustrated by the following two degenerate cases :

1. if $\sigma \to \infty$, the conditional expectation degenerates into one global linear prediction model :

$$\hat{\mathbf{y}}(x) = \frac{1}{K}\sum_i \hat{\mathbf{y}}^{(i)}(\mathbf{x}) = \mathbf{b} + \mathbf{A}\,\mathbf{x} \tag{4.60}$$

for some vector $\mathbf{b}$ and some matrix $\mathbf{A}$.

2. if $\sigma \to 0$, the "closest" observation $\mathbf{x_i}$ is choosen, i.e. let for each $\mathbf{x}$ the index $\mathrm{I}(\mathbf{x})$ be choosen in such that for $i = \mathrm{I}(\mathbf{x})$ the "distance" $\|\mathbf{x}-\mathbf{x_i}\|^2_{\mathbf{CF}^{(i)}}$ is minimal, then:

$$\hat{\mathbf{y}}(x) = \widehat{\mathbf{y}}^{(\mathrm{I}(\mathbf{x}))}(\mathbf{x}) = \mathbf{b}^{(\mathrm{I}(\mathbf{x}))} + \mathbf{A}^{(\mathrm{I}(\mathbf{x}))}\,\mathbf{x} \tag{4.61}$$

for some vectors $\mathbf{b}^{(i)}$ and some matrices $\mathbf{A}^{(i)}$.

Example 3 *The intuition behind local linear prediction is visualized in Fig.4-14. Grey line represents the idealized functional relationship $y = f(x)$. Suppose, we would like to construct the most probable forecast of y^* given x^* based on only two data points lying along the line (blue and red point). First, bell-shaped kernels are located, with centers in x_1 and x_2 (for clarity they are not plotted in the Figure). The projection (or scaled summation) of these are marginal local Parzen densities $f_x^{(1)}(x)$ and $f_x^{(2)}(x)$ centered on x_1 and x_2 . Afterwards, we calculate local covariances and create local linear prediction models $\widehat{y}^{(1)}(x)$ and $\widehat{y}^{(2)}(x)$. Then, from newly observed input x^* a line parallel to y axis is drawn and cross-points ($m^{(1)}_{Y|X=x^*}$ and $m^{(2)}_{Y|X=x^*}$) with local linear models are found. They represent mass centers of the local conditional densities*

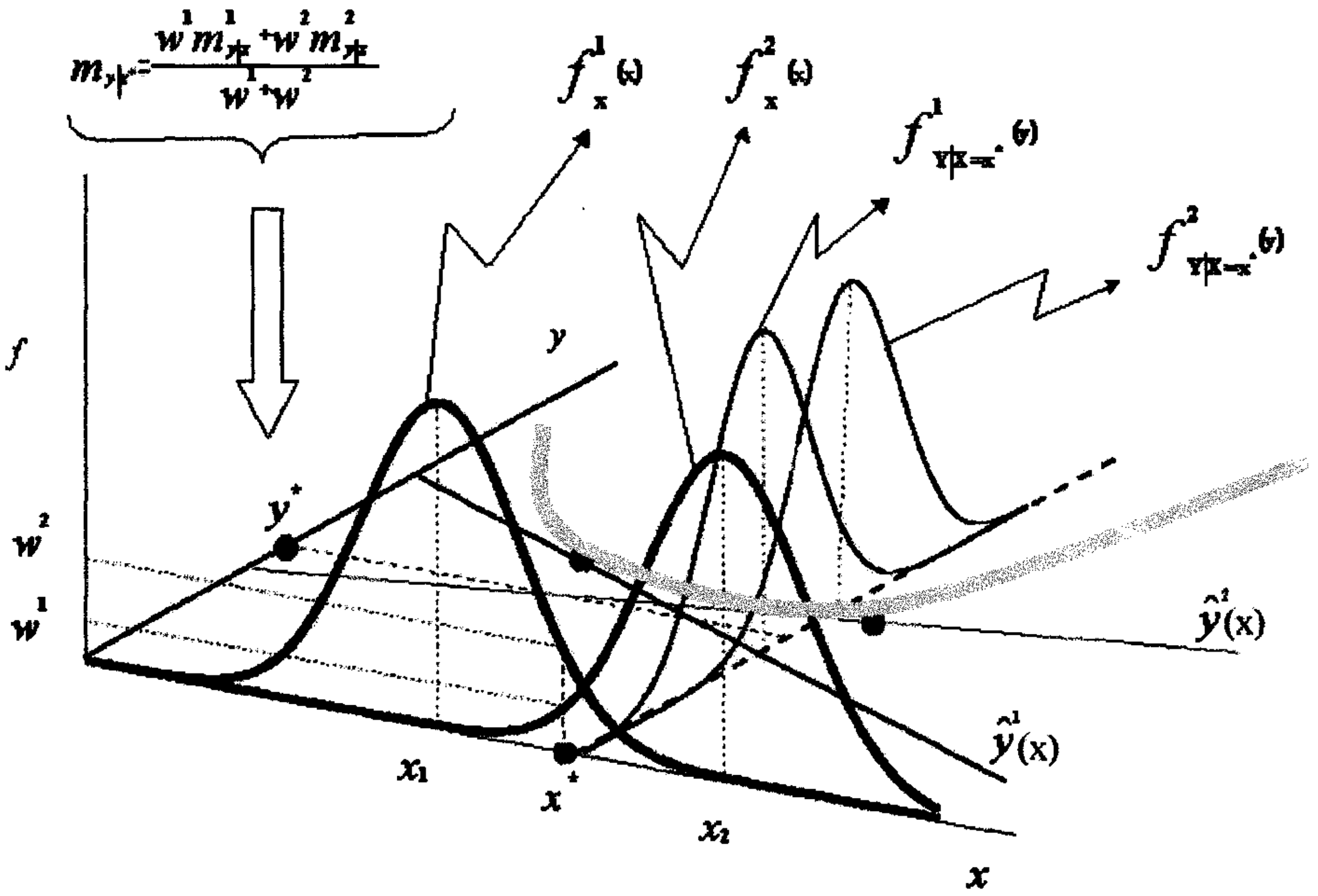

Figure 4-14: Spaces in local generalized regression.

$f^{(1)}_{Y|X=x^*}(y)$ and $f^{(2)}_{Y|X=x^*}(y)$. The final prediction $m_{Y|X=x^*}$ is weighted average of these local conditional means. The weights w_1 and w_2 originate from the marginal densities and control the local importance of each linear model.

Prediction based on locally estimated covariances leads to different interpolation - extrapolation properties than in the global Parzen models' case. Using locally linear models says nothing about the global smoothness because there is no constraint that data points have to lie near each other, so this algorithm can handle the combination of smoothness and discontinuity. In Fig.4-15, at the left most part, there is an obvious discontinuity of the data. One can notice that this discontinuity is better approximated by LC model since the sudden jump is much sharper. On the contrary, the SS model oversmooths the discontinuity. The interesting effect can be observed in the end of the data series. When we continue with the prediction beyond available data range, the SS model immediately approaches to the overall mean while in the LC case the decreasing tendency in data is extrapolated for a moment to finally reach the sample average. This is very important in applications where one deals with high dimensional input

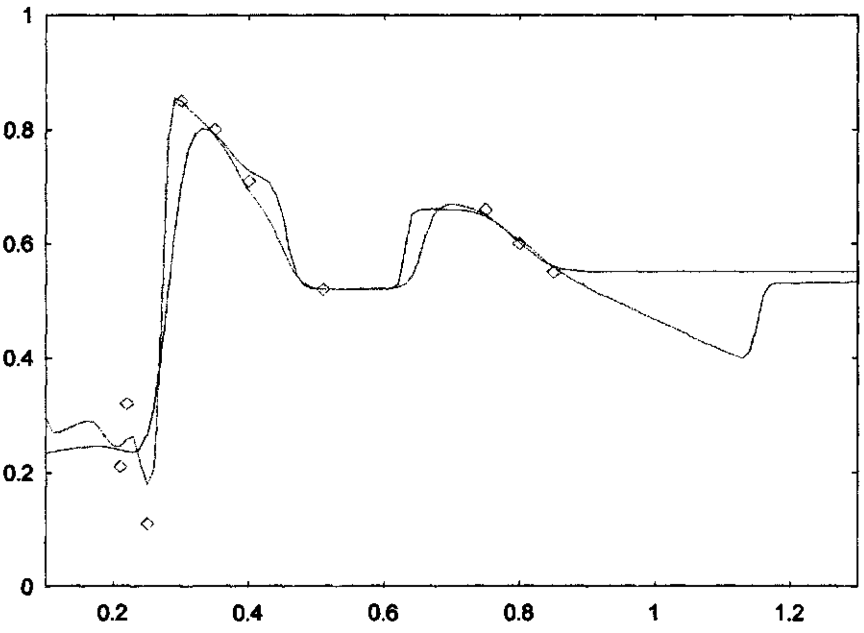

Figure 4-15: Interpolation - extrapolation properties of kernel regression. A comparison between SS (green line) and LC (blue line) models.

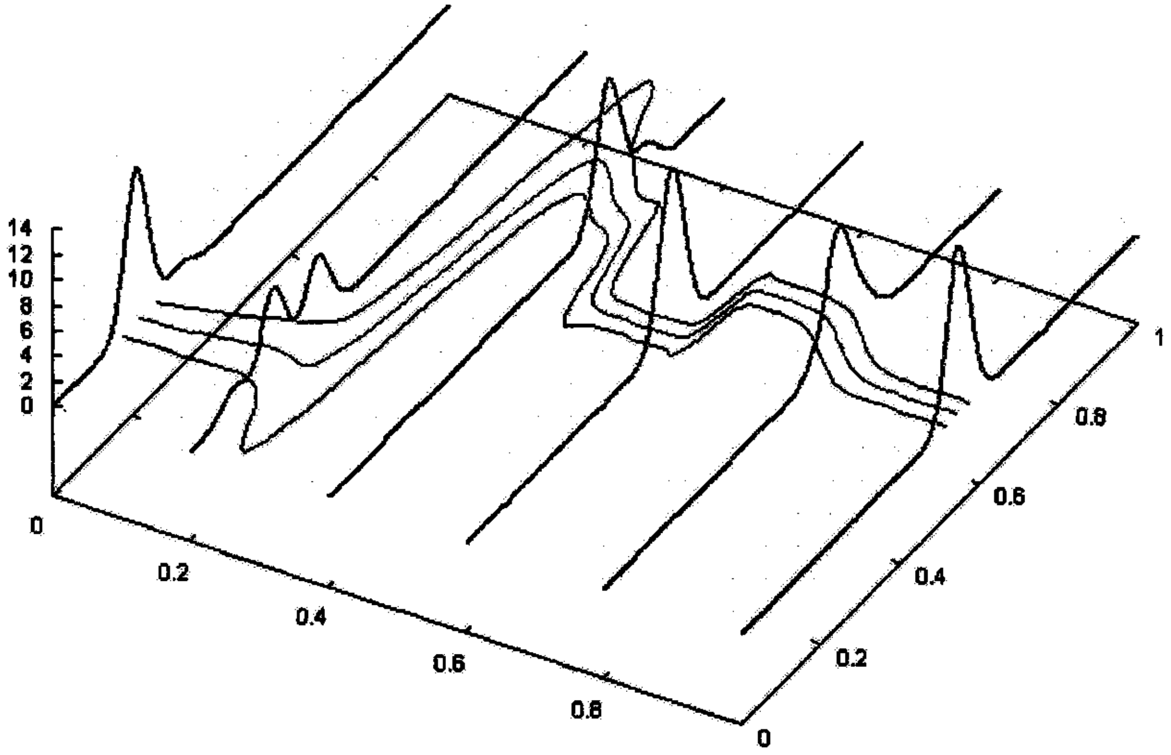

Figure 4-16: Conditional densities may happen to be multimodal.

space. Since in the high dimensions the data tends to concentrate in the edges, the whole interior of the space might be empty or locally empty (Gershenfeld [25] p.149-150). To deal with this kind of "holes" one needs the algorithm that is capable to interpolate between them. And it is exactly what local Parzen regression does.

To finish the discussion about properties of kernel regression, one more aspect is worth pointing out: multimodality of conditional distributions. This is best shown in Fig.4-16 where apart from prediction (red line) and error bands (black lines) conditional densities (blue lines) are plotted. When discontinuity occurs our algorithm has to make a decision what is the most likely prediction in the transition place. The obvious answer is : the mean. Probability density from which this mean comes from, however, has two bumps (modes) that reflect the influence of two clusters of points (Fig. 4-15).

4.2.4 Bandwidth selection schemes

Recalling the Equation 4.57 it is easy to notice that the quality of Parzen regression is dependent upon only one parameter - the bandwidth σ. In literature (see e.g. Silverman [53]) there has been a vast number of methods proposed to make choice of this parameter optimal. The optimality criteria can be grouped in two categories:

1. distribution quality methods
2. prediction quality methods

The first branch of techniques aims at optimal reconstruction of underlying probability density of the data, while the second tries to select the bandwidth that minimizes an error measure of the predictions based on Parzen regression. In this study the most commonly used criteria are considered:

1. minimum entropy (or maximum likelihood) (category 1)
2. least squares (category 2)

The first criterion was used to estimate mutual information function (see Section 4.70 and 6.3.2), while the second one was applied to calculate discharge forecasts (see Section 6.4.1).

The Entropy criterion For an arbitrary continuous density function $f(\mathbf{x})$ one defines the entropy by :

$$\begin{aligned} \mathcal{E}(f) &= -\int f(\mathbf{x}) \ln(f(\mathbf{x}))\, d\mathbf{x} && (4.62) \\ &= -\mathrm{E}[\ln(f(\mathbf{X}))] && (4.63) \end{aligned}$$

where $\mathbf{X}$ is a stochast with density f. The higher this value, the more uniform the probability distribution is, the lower the value, the more concentrated.

We could now choose the bandwidth such as to minimize the entropy of the Parzen density. This however would result in a trivial bandwidth 0, as it gives peaks in each observation point. Therefore, we eliminate the observation point self in the evaluation of $\log(f)$ in that point.

Define:

$$f_{-k}(\mathbf{x};\boldsymbol{\sigma}) = \frac{1}{(K-1)\sqrt{(2\pi\sigma^2)^N}} \sum_{i \neq k} \frac{1}{\sqrt{|\mathbf{C}^{(\mathbf{i})}|}} \exp\left[-\frac{(\mathbf{x}-\mathbf{m}^{(\mathbf{i})})^\top \mathbf{C}^{(\mathbf{i})^{-1}}(\mathbf{x}-\mathbf{m}^{(\mathbf{i})})}{2\sigma^2}\right] \tag{4.64}$$

then

$$-\mathcal{E}(f) \simeq M_{ent}(\sigma) = \sum_{k=1}^{K} \ln\left(f_{-k}(\mathbf{x}_k;\sigma)\right) \tag{4.65}$$

The optimal bandwidth is then that σ for which $M(\sigma)$ is minimal.

The least squares criterion If there are two components $\mathbf{X}$ and $\mathbf{Y}$, and corresponding observations $(\mathbf{x_1},\mathbf{y_1}),\ldots(\mathbf{x_K},\mathbf{y_K})$ another criterium based on prediction quality can be formulated :

$$M_{fit}(\sigma) = \sum \|\mathbf{y_k} - \mathrm{E}_\sigma[\mathbf{Y}|\mathbf{X} = \mathbf{x_k}]\|^2 \tag{4.66}$$

i.e. choose the bandwidth so that conditional expectation calculated with the Parzen probability with bandwidth σ gives for the data points $\mathbf{x_k}$ conditional expectations that lie as close as possible (in the least square sense) to the corresponding observations $\mathbf{y_k}$.

$$\mathrm{E}_\sigma[\mathbf{Y}|\mathbf{X}=\mathbf{x}]) = \frac{\sum_i f_{\mathbf{X}}^{(i)}(\mathbf{x};\sigma)\ \left\{\mathbf{y_i} + \mathbf{CF}_{\mathbf{YX}}^{(\mathbf{i})}\mathbf{CF}_{\mathbf{XX}}^{(\mathbf{i})^{-1}}(\mathbf{x}-\mathbf{x_i})\right\}}{\sum_i f_{\mathbf{X}}^{(i)}(\mathbf{x};\sigma)} \tag{4.67}$$

We could now choose the bandwidth such as to minimize the sum above for the Parzen density. This again would result in a trivial bandwidth 0, as the Parzen densities degenerate to dirac distributions for bandwidths going to zero in each observation point : $\mathrm{E}_0[\mathbf{Y}|\mathbf{X} = \mathbf{x_k}] = \mathbf{y_k}$. Therefore, we eliminate the observation point $\mathbf{x_k}$ self for predicting the corresponding y-value $\mathbf{y_k}$ in that point and we minimize:

$$\begin{aligned} \widehat{\mathbf{Y}}_{-k}(\mathbf{x};\sigma) &= \frac{\sum_{i\neq k} f_{\mathbf{X}}^{(i)}(\mathbf{x};\boldsymbol{\sigma})\ \left\{\mathbf{y_i} + \mathbf{C}_{\mathbf{YX}}^{(\mathbf{i})}\mathbf{C}_{\mathbf{XX}}^{(\mathbf{i})^{-1}}(\mathbf{x}-\mathbf{x_i})\right\}}{\sum_{i\neq k} f_{\mathbf{X}}^{(i)}(\mathbf{x};\boldsymbol{\sigma})} \\ M_{fit}(\sigma) &= \sum_k \left\|\mathbf{y_k} - \widehat{\mathbf{Y}}_{-k}(\mathbf{x_k};\sigma)\right\|^2 \end{aligned} \tag{4.68}$$

4.3 Mutual information as a replacement of linear correlation

The Parzen densities can be easily used for simulation (more details are given in Appendix C). With the help of Monte Carlo integration this can be applied to calculate the expectation of an arbitrary variable. In this section it will be demonstrated how to estimate some information theoretic measures with this methodology.

4.3.1 Mutual information

In hydrological time series analysis it is often the case that we want to determine how dependent are the measurements of a particular variable $s(t)$ sampled at some time to the measurements sampled at some other time $s(t+T)$. Classical approach to tackle this problem is to estimate the linear autocorrelation function of the series. There are, however, two main drawbacks with this approach:

- autocorrelation function is a linear measure of linear dependence which cannot describe potential hidden non-linear correlations in data
- there is no non-linear equivalent of autocorrelation function to estimate an amount of non linear information present in data

Hence, it is postulated here, to use a general measure of stochastic dependence of time series called *mutual information function.* This concept is derived from information theoretic (Shanon, Weaver [51]) notions of entropy of probability density function (see Eq.(4.62)). In general, mutual information $I(\mathbf{x},\mathbf{y})$ measures the reduction in uncertainty of variable $\mathbf{y}$ due to knowledge of the variable $\mathbf{x}$. The uncertainty of a distribution is made precise using the formula for entropy $\mathcal{E}(f)$. If we let $\mathcal{E}(f_{\mathbf{y}})$ be the uncertainty of $f(\mathbf{y})$ and $\mathcal{E}(f_{\mathbf{y}|\mathbf{x}})$ be the uncertainty of the conditional density $f(\mathbf{y}|\mathbf{x})$, then:

$$\begin{aligned} I(\mathbf{X},\mathbf{Y}) &= \mathcal{E}(f_{\mathbf{y}}) - \mathcal{E}(f_{\mathbf{y}|\mathbf{x}}) = \qquad (4.69) \\ &= -\sum_{\mathbf{y}} f_{\mathbf{y}}(\mathbf{y}) \log f_{\mathbf{y}}(\mathbf{y}) + \sum_{\mathbf{x}} \sum_{\mathbf{y}} f_{\mathbf{x},\mathbf{y}}(\mathbf{x},\mathbf{y}) \log f_{\mathbf{y}|\mathbf{x}}(\mathbf{y}|\mathbf{x}) \end{aligned}$$

With identity $f_{\mathbf{x},\mathbf{y}}(\mathbf{x},\mathbf{y}) = f_{\mathbf{y}|\mathbf{x}}(\mathbf{y}|\mathbf{x})f_{\mathbf{x}}(\mathbf{x})$, the expression for mutual information can also be written as:

$$I(\mathbf{X},\mathbf{Y}) = \sum_{\mathbf{x}}\sum_{\mathbf{y}} f_{\mathbf{x},\mathbf{y}}(\mathbf{x},\mathbf{y}) \log \frac{f_{\mathbf{x},\mathbf{y}}(\mathbf{x},\mathbf{y})}{f_{\mathbf{x}}(\mathbf{x})f_{\mathbf{y}}(\mathbf{y})} \tag{4.70}$$

The interpretation of this symmetric formula is straightforward : when $\mathbf{x}$ and $\mathbf{y}$ are stochastically independent, then $f_{\mathbf{x},\mathbf{y}}(\mathbf{x},\mathbf{y}) = f_{\mathbf{x}}(\mathbf{x})f_{\mathbf{y}}(\mathbf{y})$, causing the fraction in Eq.(4.70) to equal one, and thus the value of $I(\mathbf{X},\mathbf{Y})$ is zero. The value of $I(\mathbf{X},\mathbf{Y})$ grows as $\mathbf{X}$ and $\mathbf{Y}$ become more dependent. The more dependent $\mathbf{Y}$ is on $\mathbf{X}$, the more information one gains about $\mathbf{Y}$ once $\mathbf{X}$ is known, and therefore the less uncertain $\mathbf{Y}$ is when $\mathbf{X}$ is known. If one uses logarithm base 2 to estimate mutual information, the $I(\mathbf{X},\mathbf{Y})$ is expressed in bits.

In the context of time series analysis we may substitute $\mathbf{X}$ $=s(t)$ and $\mathbf{Y}$ $=s(t+T)$. Then the average mutual information between these two measurements, that is, the amount (in bits) of learned by measurements of $s(t)$ through measurements of $s(t+T)$ is:

$$I(T) = \sum_{s(t)}\sum_{s(t+T)} f_{s(t),s(t+T)}(s(t),s(t+T)) \log_2 \frac{f_{s(t),s(t+T)}(s(t),s(t+T))}{f_{s(t)}(s(t))f_{s(t+T)}(s(t+T))} \tag{4.71}$$

By general arguments given by Gallanger [22] $I(T) \geq 0$. $I(T=0)$ is directly related to the entropy of $f_{s(t)}$.

4.3.2 Linear mutual information

In special case, when $f_{s(t),s(t+T)}(s(t),s(t+T))$ is premised to be zero-mean 2-D Gaussian, the *linear mutual information* (see Fraser et al. [20]) can be easily computed from:

$$IL(T) = \frac{1}{2}\sum_{i}\log\left(\mathbf{C}_{ii}\right) - \frac{1}{2}\sum_{i}\log(d_i^2) \tag{4.72}$$

where $\mathbf{C}_{ii}$ and d_i^2 are respectively diagonal elements (variances) and eigenvalues of the 2×2 covariance matrix $\mathbf{C}$. If the above formula is evaluated using the correlation matrix instead of the variance matrix, then particularly $\mathbf{C}_{ii} = 1$ for every i, and we obtain:

$$IL(T) = -\frac{1}{2}\sum_{i}\log(d_i^2) \tag{4.73}$$

The linear redundancy, according to its definition, reflects linear dependence structures contained in the correlation matrix **C** of variables under study. It was proposed by Palus [44] to compare classical mutual information with linear mutual information in order to check whether the $\mathcal{LG}$ description of the particular process is sufficient. Large discrepancies between the *shapes* of the two functions suggest the existance of nonlinearities in time series. This issue will be revisited in Section 6.3.2.

4.3.3 Estimation by Monte Carlo integration

The calculation of mutual information strongly depends on probability density estimator used. The classical approach is based on histogramming. The more recent algorithms are based on Parzen density estimators. Both methods, however, suffer from a problem of discretization. To integrate Eq.(4.71) directly, the probability space has to be first split up in a number of cells and for each cell the value of the probability density is calculated. This procedure is not very effective numerically and time consuming simply because there are potentially many empty (with no data) cells in space. Yet, reffering to the discussion from page 57, for higher dimensional probability densities the number of those cells increases as a power law of dimension (the so-called curse of dimensionality). Apart from this, in high dimensional spaces there are vast amounts of data points required to get reliable estimates of mutual information. Since Eq.(4.70) can be rewritten as the expected value of the function:

$$I(\mathbf{X},\mathbf{Y}) = E\left[\log\frac{f_{\mathbf{x},\mathbf{y}}(\mathbf{x},\mathbf{y})}{f_{\mathbf{x}}(\mathbf{x})f_{\mathbf{y}}(\mathbf{y})}\right] \tag{4.74}$$

Bonnlander [11] suggested to use an unbiased estimator:

$$\widehat{I}_n(\mathbf{X},\mathbf{Y}) = \frac{1}{n}\sum_{i=1}^{n}\log\frac{f_{\mathbf{x},\mathbf{y}}(\mathbf{x}_i,\mathbf{y}_i)}{f_{\mathbf{x}}(\mathbf{x}_i)f_{\mathbf{y}}(\mathbf{y}_i)} \tag{4.75}$$

For $n \longrightarrow \infty$ the $\widehat{I}_n(\mathbf{X},\mathbf{Y}) \rightarrow I(\mathbf{X},\mathbf{Y})$. For small samples, however, it may happen that $\widehat{I}_n(\mathbf{X},\mathbf{Y}) \neq I(\mathbf{X},\mathbf{Y})$. Therefore, it is proposed here to use Monte Carlo integration of Eq.

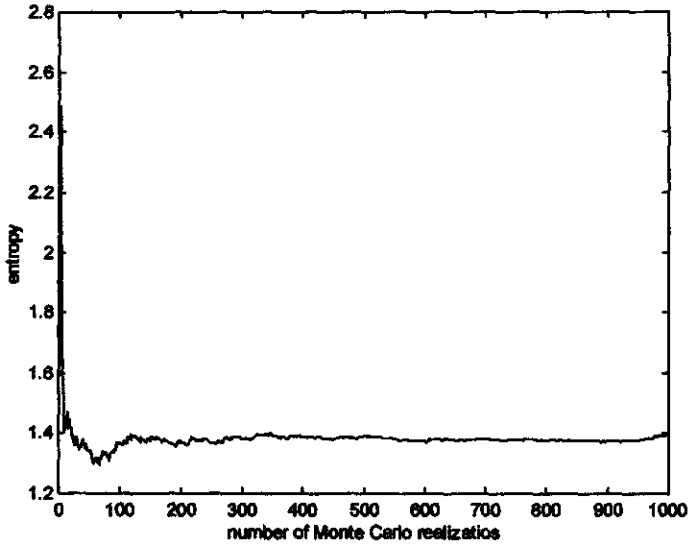

Figure 4-17: Entropy of Gaussian distribution by Monte Carlo integration.

(4.71). If $\widetilde{\mathbf{x}}_1, \ldots, \widetilde{\mathbf{x}}_n$ and $\widetilde{\mathbf{y}}_1, \ldots, \widetilde{\mathbf{y}}_n$ are independent realizations drawn from probability density $f_{\mathbf{x},\mathbf{y}}(\mathbf{x}, \mathbf{y})$, the mutual information can be approximated by:

$$\widehat{I}_n(\mathbf{X}, \mathbf{Y}) = \frac{1}{n} \sum_{i=1}^{n} \log \frac{f_{\mathbf{x},\mathbf{y}}(\widetilde{\mathbf{x}}_i, \widetilde{\mathbf{y}}_i)}{f_{\mathbf{x}}(\widetilde{\mathbf{x}}_i) f_{\mathbf{y}}(\widetilde{\mathbf{y}}_i)} \tag{4.76}$$

The above Formula converges to a true value of $I(\mathbf{X}, \mathbf{Y})$ as a function of n (Fig.4-17 gives an example of a speed of convergence).

Example 4 *To show how the Monte - Carlo integration works in practice, the entropy of Gaussian:*

$$f(x) = \frac{1}{\sqrt{2\pi s_x^2}} \ \exp\left[-\frac{(x - m_x)^2)}{2s_x^2}\right] \tag{4.77}$$

with $m_x = 0$ and $s_x^2 = 1$ was estimated. In this case we can calculate entropy analytically, being :

$$\mathcal{E}(f) = -\int_{-\infty}^{\infty} f(x) \ \ln(f(x)) \, dx = 1.419 \tag{4.78}$$

The results of Monte Carlo integration are shown in Fig.4-17. Within 1000 iterations a good approximation of the entropy was found.

A real- life example follows on page 106.

Chapter 5

Non-linear time series models

The goal of this Chapter is to present the ideas concerning non- linear time series models that have been invented during the last few years. The methods reviewed here can be grouped in two categories: local models and global models. Those schemes can be viewed as a generalization of the well-known linear autoregression approach to time series modelling. Section 5.1 introduces the concept of the state space reconstruction of a deterministic system based on single observable (output variable). This analysis provides one with the estimates of global and local dimensions of the dynamics. In the context of forecasting methods outlined in Section 5.2 , these dimensions can be interpreted as number of input variables to global and local models respectively.

5.1 Exploring the state space

Although the tools described in the next sections originate from deterministic chaos theory the author's intent was not to introduce them as chaos - detection methods but as an excellent complementary source of information for standard linear time series algorithms, which should be included in arsenal of every contemporary signal analyst. Unlike in Section 2, it was easier here to start the considerations with an example of continuous-time system.

5.1.1 Attractor reconstruction problem

The state space of a dynamical system at any time can be specified by a state - space vector where the coordinates of the vector are independent degrees of freedom of the system. Generally

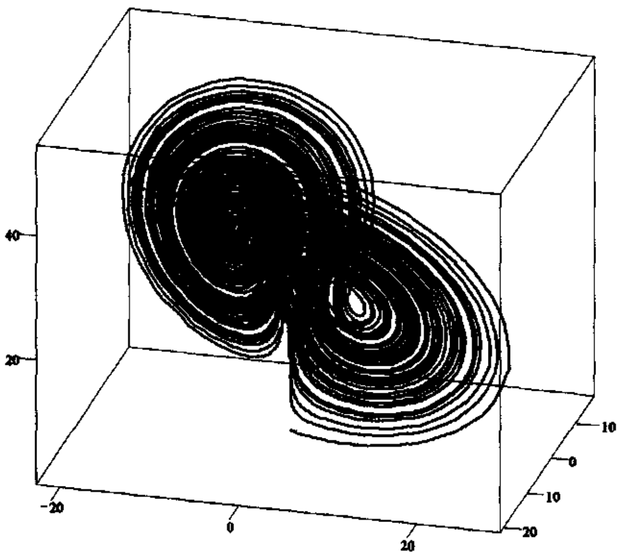

Figure 5-1: Lorenz attractor. In the Runge - Kutta integration routine a time step $\tau_s = 0.01$ was used.

the number of ordinary first order differential equations describing the system determines the number of active degrees of freedom. For the sake of example in Fig. 5-1 a trajectory generated by classical Lorenz system [36] is shown. The equations describing this system read:

$$\begin{aligned} \dot{x_1}(t) &= \psi(x_2 - x_1) \\ \dot{x_1}(t) &= -x_1 x_3 + r x_1 - x_2 \\ \dot{x_3}(t) &= x_1 x_2 - b x_3 \end{aligned} \tag{5.1}$$

and we use standard parameter values $\psi = 16$, $b = 4$, $r = 45.92$. The state vector trajectory is given by $\mathbf{x}(t)= [x_1(t), x_2(t), x_3(t)]$ at any time $t = t_0 + n\tau_s$, where τ_s is a sampling interval. Of practical importance is, however, the question whether by measuring just one variable ($x_1(t)$ or $x_2(t)$ or $x_3(t)$) we can say something about the dynamics of the whole system, i.e. whether we can reconstruct the system's state space and, as a consequence, make predictions of the future. This is the "embedding theorem" formulated by Takens [58] which gives the positive answer to the above problem. It should be clearly stated, however, that the state space reconstruction

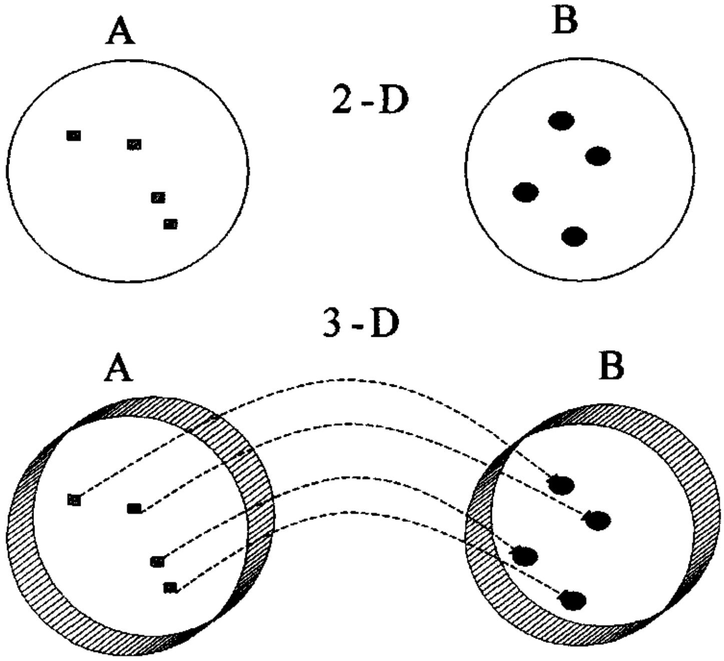

Figure 5-2: Simple embedding operation.

does not aim at finding underlying differential equations to predict the future behaviour of a system. In contrast, the purpose of the reconstruction is to reveal geometrical information about the state space hidden in one of the observed variables. Out of this geometry, the predictions are constructed.

Before technical description of the essence of the Takens' theorem starts, it would be useful to build up the intuition of what is actually meant by the embedding operation. In upper part of Figure 5-2 we can see two sets A and B containing some elements. The problem is how to connect these elements using continuous paths or how to build a mapping $A \longrightarrow B$ without intersecting boundaries of the sets. In 2-D space this is impossible, however, going into the space that is one dimension higher can do the trick (lower part of Figure 5-2). This topological procedure is designated as the embedding. So by adding an extra degree of freedom (3rd dimension in this case) one is allowed to perform operations that are not possible in the space of lower dimension.

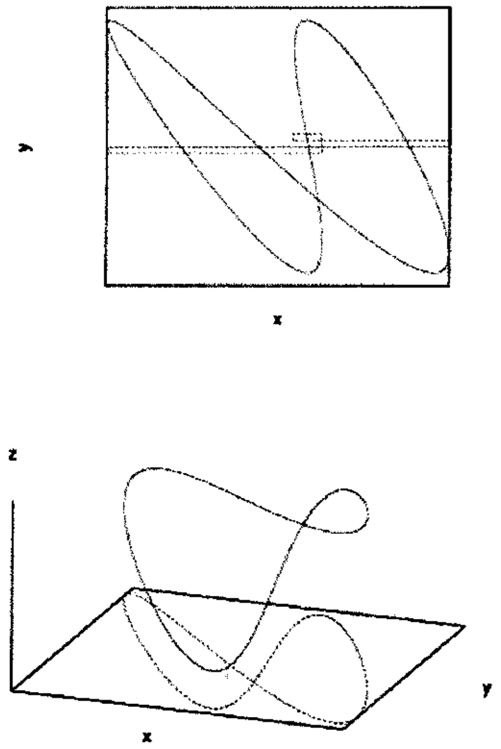

Figure 5-3: Projection of one dimensional manifold in different dimensions.

5.1.2 Takens theorem

The embedding theorem states that if, in the original state space, system produces trajectories, which, after transients are gone, lie on a geometric object (called a dynamical attractor of dimension[1] d_A), then the object can be unambiguously seen without any intersections of the orbits in another space of integer dimension $d_E > 2d_A$, or larger, comprised of coordinates that are (almost) arbitrary non-linear transformations of original state space coordinates. The absence of intersections, in the latter space means that trajectory is uniquely resolved when d_E is large enough. Overlaps of the trajectory may occur in lower dimensions which automatically violates determinism[2] and in consequence possibility of making forecasts.

[1] The attractor for which d_A takes non-integer value is called *starnge attractor.*

[2] Determinism in the Laplacian sense implies that each point in the state space has only one possible future.

Almost any set of d_E coordinates is equivalent by the embedding theorem. Each set is a different way of unfolding the attractor from its projection onto the observations. Formally an autonomous system producing trajectories $\mathbf{x}(t) = [x_1(t), x_2(t), ..., x_n(t)]$ through the dynamics is given by:

$$\dot{\mathbf{x}}(t) = \mathbf{F}(\mathbf{x}(t)) \tag{5.2}$$

The output, typically one dimensional signal, is expressed as:

$$s(t) = h(\mathbf{x}(t)) \tag{5.3}$$

With mild restriction of the functions $\mathbf{F}(\mathbf{x})$ and $h(\mathbf{x})$ (for a broad discussion see e.g. Abarbanel [1]), any independent set of quantities related to $s(t)$ can serve as the coordinates for the reconstructed state space. For instance time derivatives of $s(t)$ are the natural set of independent coordinates. The first and second derivative of $s(t)$, provided that τ_s is very small, can be approximated by:

$$\dot{s}(t) \approx \frac{s(t + (n+1)\tau_s) - s(t + n\tau_s)}{\tau_s} \tag{5.4}$$

$$\ddot{s}(t) \approx \frac{s(t + 2\tau_s) - 2s(t + \tau_s) + s(t)}{2\tau_s^2} \tag{5.5}$$

Similar approximations can be constructed for higher order derivatives as well. However, one has to remember that this kind of differencing is also a high-pass filtering operation which puts an emphasis on instrument errors and noise contained in real data. On the other hand, one can easily observe that the time delay values of $s(t)$ are new information that enters each derivative approximation. So, they can be used as alternative coordinate set for the state space reconstruction. This way one also avoids computations on observations themselves. In those new coordinates the following vectors are formed:

$$\mathbf{y}(t) = [s(t), s(t - T\tau_s), s(t - 2T\tau_s), ..., s(t - (d_E - 1)T\tau_s)] \tag{5.6}$$

where T and d_E are integer numbers referred to as reconstruction time lag and embedding dimension respectively. An example of this reconstruction is shown in Fig.5-3 adopted from Kanz et al. [32] where embedding of one dimensional manifold in two and three dimensions is shown. One can notice that only in three dimensions is self intersection avoided (in the top picture overlapping regions in one dimension are presented schematically as dotted lines). Hydrological examples of the atrractors reconstructed by the embedding procedure are presented in Fig. 5-4 adapted from Frison et al. [21]. In their extensive study on characterizing and classifying ocean waterlevels, several time series from open coast (upper three figures) and Chesapeake Bay (middle three figures + leftmost bottom figure) tide stations were analyzed. The differences between the dynamics of waterlevels from open coast stations and those at the mouth of Chesapeake Bay are evident. For comparison the attractors of correlated noise and

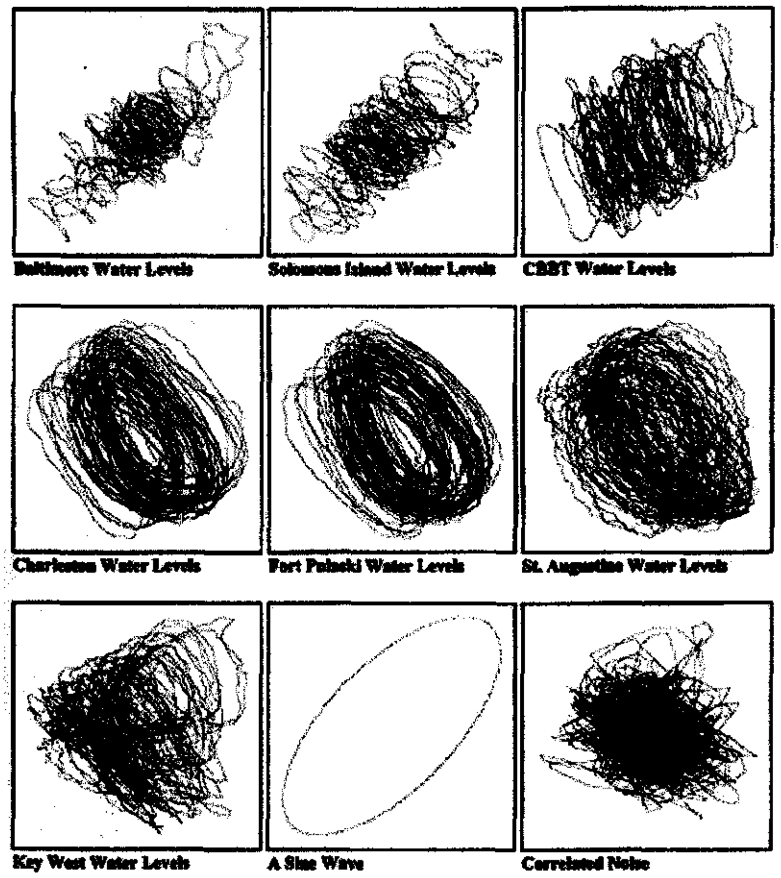

Figure 5-4: The reconstructed waterlevel attractors.

fora a sine wave (one periodic component) are also shown. This gives a brief overview of the hydrological applicability of the state - space reconstruction concepts.

When one looks again at the Formula 5.6 there are two parameters which the reconstruction is dependent upon, namely T and d_E. In theory [58] the choice of T value is not so important provided that one deals with infinite amount of infinitely clean data. On the other hand this choice may be crucial when real-life, limited and often noisy, data sets are concerned. In the literature there is no clear algorithm to estimate the proper T , however, there is some kind of prescription that proved to work well in practice. It will be briefly discussed in the next section.

The proper estimation of the second parameter d_E is of an immense practical importance as well. From the geometrical point of view, attractor structure will not be fully unfolded unless one goes to dimension as high as d_E . This is the global dimension that reveals geometrical properties of the dynamics and moreover allows one to construct global prediction algorithms. Section 5.1.4 addresses the d_E estimation issues.

5.1.3 Time delay by mutual information

Fraser and Swinney [20] suggested to use the first minimum of the mutual information function (see Section 4.70) as an optimal time delay for state space reconstruction. However, this prescription is found inappropriate (see Section 6.3.2) since mutual information may not show any local minimum at all, or spurious local minima can occur due to instabilities in estimation procedures. The safest way out in this case is to take $T = 1$ as it was done in this study.

5.1.4 Estimating the embedding dimension

Let us assume that one makes a state space reconstruction in dimension d :

$$\mathbf{y}(t) = [s(t), s(t-T), s(t-2T), ..., s(t-(d-1)T)] \tag{5.7}$$

Examine the nearest neighbor of the vector $\mathbf{y}(t)$ in sate space with time label $\bar{t}$ namely the vector:

$$\mathbf{y}^{NN}(\bar{t}) = [s^{NN}(\bar{t}), s^{NN}(\bar{t} -T), s^{NN}(\bar{t} -2T), ..., s^{NN}(\bar{t} -(d-1)T)] \tag{5.8}$$

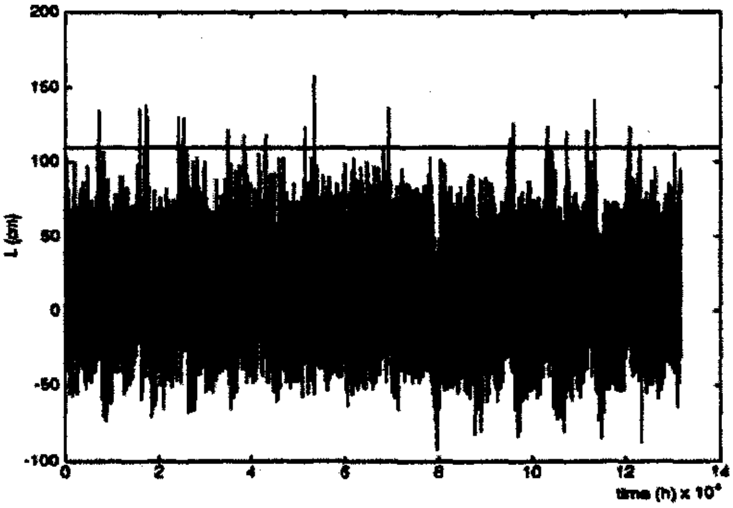

Figure 5-5: Signal trace of waterlevel at Venice Lagoon (reproduced from Zaldivar et al. [66]).

If the vector$\mathbf{y}^{NN}(\bar{t})$ is a **true** neighbor of $\mathbf{y}(t)$ it came to the neighborhood of $\mathbf{y}(t)$ through dynamical origins. If $\mathbf{y}^{NN}(\bar{t})$ is a **false** neighbor it has arrived into the neighborhood by projection from higher dimension because the present dimension d does not unfold the attractor. Then by going into dimension $d+1$ one may move this false neighbor out of the neighborhood of $\mathbf{y}(t)$. By comparing the distance between the vectors $\mathbf{y}^{NN}(\bar{t})$ and $\mathbf{y}(t)$ in dimension d with the distance between additional components of these vectors in $d+1$ which are just $s^{NN}(\bar{t}\ -dT)$ and $s(t-dT)$, one can easily identify true and false neighbors. In other words, if the additional distance is large compared to the distance in dimension d between nearest neighbors, one has a false neighbor. If it is not large one has a true neighbor.

To put the above discussion into more formal framework, let us define the Euclidean distance between the nearest neighbor points as seen in dimension d is :

$$R_d(t)^2 = \sum_{m=1}^{d} \left[s^{NN}(\bar{t}\ -(m-1)T) - s(t-(m-1)T) \right]^2 \tag{5.9}$$

while in dimension $d+1$ it is:

$$R_{d+1}(t)^2 = R_d(t)^2 + \left| s^{NN}(\bar{t}\ -dT) - s(t-dT) \right|^2 . \tag{5.10}$$

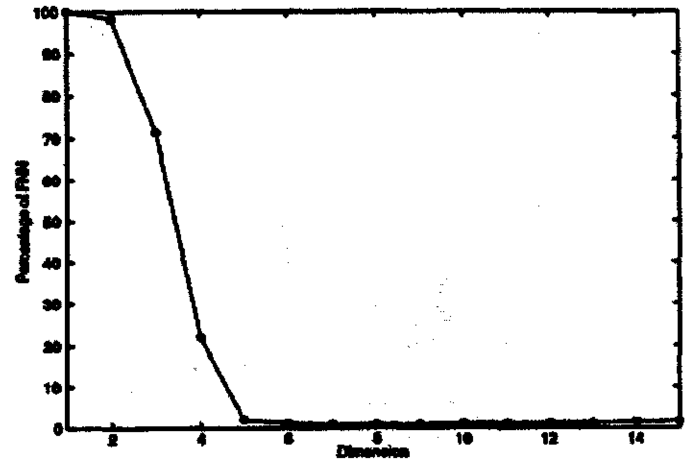

Figure 5-6: Embedding dimension calculation for Venice Lagoon data (adapted from Zaldivar et al. [66]).

The distance between points when seen in dimension d+1 relative to the distance in dimension d is:

$$\sqrt{\frac{R_{d+1}(t)^2 - R_d(t)^2}{R_d(t)^2}} = \frac{\left|s^{NN}(\bar{t}\ -dT) - s(t - dT)\right|}{R_d(t)} \tag{5.11}$$

When this quantity is larger then some threshold we have a false neighbor of **type 1** in dimension d. Frison et al.[21] proposes this number to be about 0.2 or so.

There is a subtle issue associated with searching for the false neighbors. As mentioned earlier, when we go to the higher dimensional spaces the data will be crowded to the edges of the state space. This is due to the fact that volume of space is proportional to distance to the power of dimension (Gershenfeld [25] p.149–150). Then no near neighbors could have been classified as true neighbors just because of curse-of-dimensionality effects. Second criterion needs to be formulated to tackle this problem :

$$\frac{\left|s^{NN}(\bar{t}\ -dT) - s(t - dT)\right|}{R_A} \tag{5.12}$$

where R_A is called in physical literature as a nominal "radius" of attractor and is nothing else than standard deviation of data set under analysis. According to Abarbanel [1] if the criterion

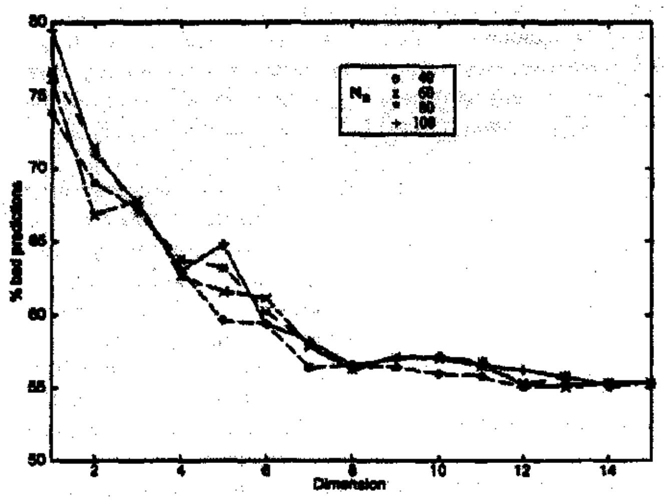

Figure 5-7: Local dynamical dimension analysis fror Venice Lagoon waterlevels (adapted from Zaldivar et al. [66]).

5.12 is greater then number of order of 2 one has a false neighbor of **type 2.**

The actual goal of the above false nearest neighbor search (FNN) is to find a dimension d_E where the sum of FNN's of the two types drops off to 0 (or near zero for real-life data) and stays there for $d > d_E$. To illustrate this concept the results of the FNN analysis performed on sea waterlevel signal (Fig. 5-5) by Zaldivar et al. [66] are presented. The effect of the embedding dimension calculation is plotted in Fig.5-6. In this case $d_E = 5$ or 6. The results suggest that, in general, global models for classification and prediction of the waterlevel signal can be made in 5 or 6-D space. Additionally, as Frison et al. [21] reported, the FNN analysis is a robust procedure to small data sets.

5.1.5 Local dynamical dimension

The embedding theorem is concerned only with global properties of the dynamics. It provides the information on how many degrees of freedom, or input variables should be used for *globally* parametrized models. The question remains about possibility of reducing the active number of inputs for *local* prediction models. Word "local" signifies that model parameters are estimated locally in the state space. By exploring small regions of the attractor the short term prediction of a system's behavior can be defined.

The problem of finding local dynamical dimension d_L can be rephrased by asking what subspace of dimension $d_L \leq d_E$ allows one to make accurate local neighborhood to neighborhood maps on the attractor (Abarbanel [1], Abarbanel and Kennel [3]). The calculations are as follows. We choose a point $\mathbf{y}(t)$ from trajectory embedded in dimension d_E and select its N_B neighbors (where $\mathbf{y}^{(r)}(\bar{t})$ is the r^{th} neighbor of $\mathbf{y}(t)$). Then we provide a local rule how these points evolve in one time step into the same N_B points near $\mathbf{y}(t+1)$. Afterwards, the quality of the predictions is tested in $d \leq d_E$ dimensional subspaces. The dimension for which the quality of the predictions becomes independent of N_B and d, is designated as the local dynamical dimension d_L. The local coordinate system might be determined by doing local principal components analysis (PCA) [26] in d_E dimensional space. Eigenvalues of the local $d_E \times d_E$ covariance matrix defined as:

$$\mathbf{C}(t) = \frac{1}{N_B} \sum_{r=1}^{N_B} [\mathbf{y}^{(r)}(\bar{t}) - \mathbf{y}^{av}(t)][\mathbf{y}^{(r)}(\bar{t}) - \mathbf{y}^{av}(t)]^T \tag{5.13}$$

where

$$\mathbf{y}^{av}(t) = \frac{1}{N_B} \sum_{r=1}^{N_B} \mathbf{y}^{(r)}(\bar{t}) \tag{5.14}$$

are ordered in size and as the basis for d_L dimensional subspace we choose eigendirections associated with the largest eigenvalues. Once the coordinate system is established the d_E dimensional vectors $\mathbf{y}^{(r)}(\bar{t})$ are projected onto d_L dimensional space to form $\mathbf{z}^{(r)}(\bar{t})$ vectors. The evolution of $\mathbf{z}^{(r)}(\bar{t})$ over one time step is found from a local polynomial map:

$$\mathbf{z}^{(r)}(\bar{t}) \longrightarrow \mathbf{z}^{(r)}(\bar{t}+1) \tag{5.15}$$

$$\mathbf{z}^{(r)}(\bar{t}+1) = \mathbf{A} + \mathbf{B}\mathbf{z}^{(r)}(\bar{t}) + \mathbf{C}\mathbf{z}^{(r)}(\bar{t})\mathbf{z}^{(r)}(\bar{t}) + ... \tag{5.16}$$

The quantities **A**,**B**,**C** are tensors of an appropriate rank. They can be found from a least squares minimization of:

$$\sum_{r=1}^{N_B} \left| \mathbf{z}^{(r)}(\bar{t}+1) - \mathbf{A} - \mathbf{B}\mathbf{z}^{(r)}(\bar{t}) - \mathbf{C}\mathbf{z}^{(r)}(\bar{t})\mathbf{z}^{(r)}(\bar{t}) - ... \right|^2 \tag{5.17}$$

Having determined the local polynomial map one asks how well it predicts forward in time. The quality of the predictions is quantified by number of the trajectories associated with the N_B neighbors which remain within some fraction of R_A for a number of samples ζ forward in time. A bad prediction is one when the trajectory diverges from the original neighborhood by more than R_A before it reaches the prediction limit. Percentage of bad predictions is averaged over the number of neighborhoods i.e. number of $\mathbf{y}(t)$ points for which the neighborhoods are formed.

The results of the local dimension analysis for the Venice Lagoon data are illustrated in Fig.5-7 where percentage of bad predictions is plotted as a function of dimension and N_B (typically $N_B = 40, 60, 80, 100$). A good estimate of d_L here is 8, or 7 for the bold.

5.2 Prediction

The reconstructed state vectors $\mathbf{y}(t)$ can be used for modelling the dynamics of a system. This allows one to make predictions of the future trajectories based on any newly observed point in the state space. There are several approaches to this issue that will be outlined in this section. At first, local polynomial models based on local neighborhood to neighborhood maps in the state space will be described. Then we turn to examples of "from global to local" models' parametrization schemes by exploring two variations on the theme of neural networks: multilayer feedforward and generalized regression neural networks. The latter algorithm originates from parallel implementation of Parzen regression from Section 4.2.

5.2.1 Local polynomial models

The material presented in this sub-section should be treated as a complementation of the concepts described within a framework of local dynamic dimension d_L. Here, however, the emphasis will be put on the technical issue of the predictions-making itself .

To build local maps that describe evolution of neighborhood, one starts with an observed point, $\mathbf{y}(t)$, on the attractor reconstructed in dimension d_E . In a neighborhood of $\mathbf{y}(t)$ with

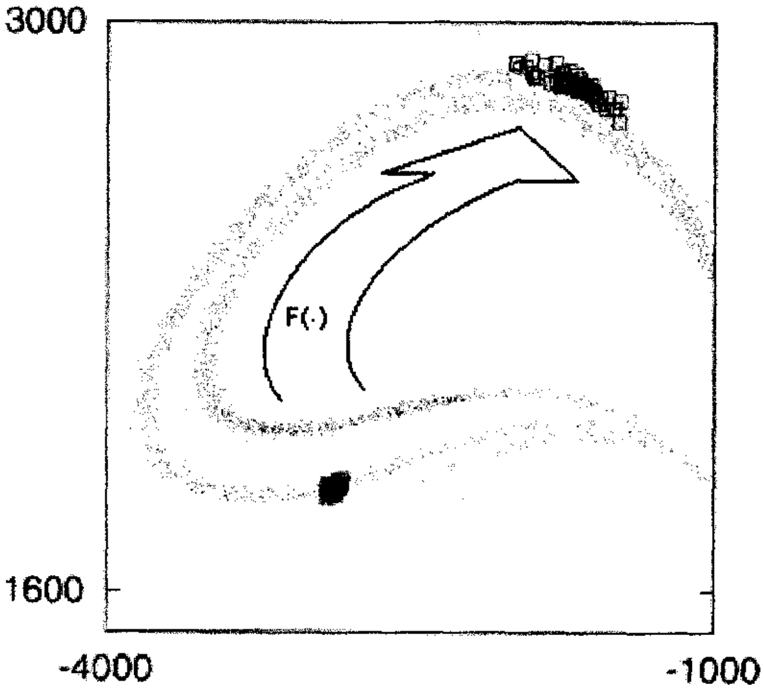

Figure 5-8: As an example of simple local prediction. Enlarged region of a state space of NMR laser data is shown (Kanz et al. [32]). For an examplary point in the state space all its true neighbours (the cluster of squares on the lower part) are highlighted and their images obtained by $\mathbf{F}^{(t)}(\cdot)$ mapping after one time step of the dyamics.

N_B neighbors $\mathbf{y}^{(r)}(\bar{t})$ where $r = 1, 2, .., N_B$, the unknown local map $\mathbf{F}^{(t)}(\cdot)$, that evolves points on the attractor $\mathbf{y}^{(r)}(\bar{t} + 1) = \mathbf{F}^{(t)}(\mathbf{y}^{(r)}(\bar{t}))$ can be expanded as:

$$\mathbf{F}^{(t)}(\mathbf{y}^{(r)}(\bar{t})) = \sum_{m=1}^{M} \mathbf{c}(\bar{t}, m)\phi_m(\mathbf{y}^{(r)}(\bar{t})) \tag{5.18}$$

in terms of M basis functions $\phi_m(\cdot)$. There are many possible choices of the $\phi_m(\cdot)$ (for example see Casdagli et al. [14]), however, one form that often works well in practice is just an $M - th$ order polynomial as introduced in Eq.(5.16). The above map determines where the neighbor point $\mathbf{y}^{(r)}(\bar{t})$ goes in unit time interval of the dynamics. By minimizing:

$$\sum_{r=1}^{N_b} \left| \mathbf{y}^{(r)}(\bar{t}\ {+1}) - \sum_{m=1}^{M} \mathbf{c}(\bar{t},m)\phi_m(\ \mathbf{y}^{(r)}(\bar{t})) \right|^2 \tag{5.19}$$

which is a well known linear problem, we get the coefficients $\mathbf{c}(\bar{t},m)$ that parametrize local neighborhood map $\mathbf{F}^{(t)}(\cdot)$ at time t. An example of neighborhood to neighborhood mapping within one step of the dynamics is presented on Fig.5-8

The above framework allows one to parametrize a local prediction model in $d_E = d_L$ dimensional space. On the other hand, if local dimension was identified to be $d_L < d_E$ the natural subspace in which to make a model is d_L dimensional. The way to proceed in this case can be summarized as follows :

- choose $\mathbf{y}^{(r)}(\bar{t})$ neighbors of $\mathbf{y}(t)$ in d_E dimensional space
- based on analysis from Section 5.1.5 select d_L dimensional subspace in which to make a model
- form d_L dimensional vectors $\mathbf{z}^{(r)}(\bar{t})$ by selecting d_L components of $\mathbf{y}^{(r)}(\bar{t})$
- construct local model that maps $\mathbf{z}^{(r)}(\bar{t}) \longrightarrow \mathbf{z}^{(r)}(\bar{t}\ {+1})$ by analogy with Eq.(5.18) and (5.19)

Once the model is parametrized (for each data point $\mathbf{y}(t)$ in the state space there is a polynomial map that evolves it into $\mathbf{y}(t+1)$) one can make predictions for any (reasonable) *time lead* ζ, sometimes also called *prediction horizon*. Suppose that we measured a new data point in time series $\mathbf{y}_{new}(t)$. Its evolution ζ steps ahead can be foreseen by first locating the nearest point in the training data, call it $\mathbf{y}(\bar{t_1})$ that provides the coefficients $\mathbf{c}(\bar{t},m)$ and secondly locating the point that follows $\mathbf{y}_{new}(t)$ after one time step is $\mathbf{y}_{new}(t+1)$

$$\mathbf{y}_{new}(t+1) = \sum_{m=1}^{M} \mathbf{c}(\bar{t_1},m)\phi_m(\mathbf{y}_{new}(t)\ ) \tag{5.20}$$

if the nearest point to $\mathbf{y}_{new}(t+1)$ in the training set is $\mathbf{y}(\bar{t_2})$ then:

$$\mathbf{y}_{new}(t+2) = \sum_{m=1}^{M} \mathbf{c}(\bar{t_2},m)\phi_m(\mathbf{y}_{new}(t+1)) \tag{5.21}$$

- One proceeds in this iterative $\mathbf{y}_{new}(t) \rightarrow \mathbf{y}_{new}(t+1) \rightarrow \mathbf{y}_{new}(t+2) \rightarrow \ldots \rightarrow \mathbf{y}_{new}(t+\zeta)$ fashion until the prediction horizon is reached.

Remark 5 ***Direct vs. iterative prediction.*** *Assume one is given a time series $x_1, \ldots, x_N$ and asked to provide continuation. We apply our method to predict one time unit ahead and get an estimate $\hat{x}_{N+1}$. To get an estimate of $\hat{x}_{N+2}$there are two obvious choices. The* ***direct prediction*** *method means that the original method is applied to $x_1, \ldots, x_N$ to predict two time units ahead. In contrast,* ***iterated prediction*** *means applying the method to $x_1, \ldots, x_N$, $\hat{x}_{N+1}$to predict one step ahead.*

The predictor described in this section is clearly an iterative method since it makes a large number of unit steps into the future of $\mathbf{y}_{new}(t)$ to reach ζ. If one builds a model that goes directly from $\mathbf{y}_{new}(t)$ to $\mathbf{y}_{new}(t+\zeta)$ more accurate interpolation methods would be required since the neighborhood one must deal with is much larger. By proceeding in smaller steps the prediction accuracy is increased since a kind of updating is done in the reconstructed state space.

5.2.2 Neural networks : algebraic and probabilistic models

In this Section neural networks will be presented as a non-linear regression tools. Unlike the state space models, neural networks do *static* input - output mappings with no internal dynamics. Since algebraic neural networks are described in details in many publications (see e.g. Bishop [9], Masters [39, 40]), only some intuitive overview of this method will be provided in what follows. More technical treatment will be given to the probabilistic networks. Finally, some pointers to the hydrological literature on neural networks will be outlined.

A basic concept

There are two views on classical linear regression :

- an algebraic one : find a number a such that for a given data set $\sum(y_i - a\,x_i)^2$ is minimal;
- a probabilistic one : given the normal probability model $y = a\,x + s\,\varepsilon$, where ε has a standard normal distribution, find an optimal a and s according to a maximum likelihood principle (or a minimal variance principle, which is the same for normal densities).

Classical multi-layer feed forward networks (MLFNNs) can be considered as generalizations of the first view above. Probabilistic Neural Networks (PNNs for short) are generalizations of the second view.

Neural networks were originally constructed to model the human brain. A brain consists of many interconnected elements called neurons. The power of the brain is due to the complexity of these connections, rather than to the internal complexity of the neurons. This is also the main characteristic of the mathematical models constructed to model these brain functions.

Algebraic MLFNNs

A neural network model consists of number of neurons or processing units each of which performs a nonlinear transformation on its inputs. To understand the working of this whole system we will consider a simple example given by Torfs and Bier [61]. Figure 5-9 illustrates the operation

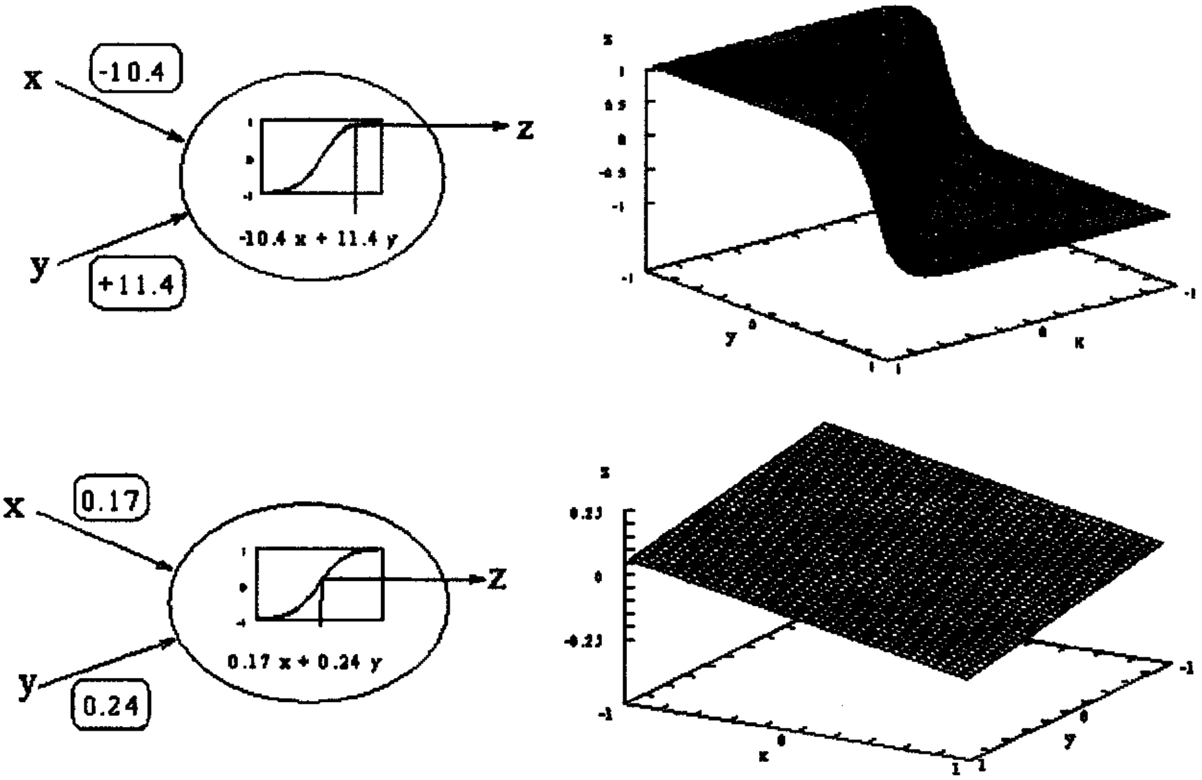

Figure 5-9: Single neuron examples

of a single neuron. In both examples there are two inputs, named x and y. In general any number of inputs is allowed. The first computational step is to calculate a weighted sum of the inputs. For the top case, the weights are -10.4 and +11.4, and the weighted sum of the inputs is then -10.4x + 11.4y. This weighted sum is then transformed by a non-linear function, often called activation function. A typical activation function is $\eta(x) = \tanh(x)$. The result of this is called the output of the neuron. The right top part of Fig.5-9 shows the surface generated by this formula. This is clearly non-linear function. Since the weights were rather high more of the " ± 1" values of the activation function were selected resulting in surface that approximates a step function. In the bottom part of the figure the weights were chosen small resulting in linear surface. So, by adjusting the neuron's weights, linear and non-linear behavior can be modeled.

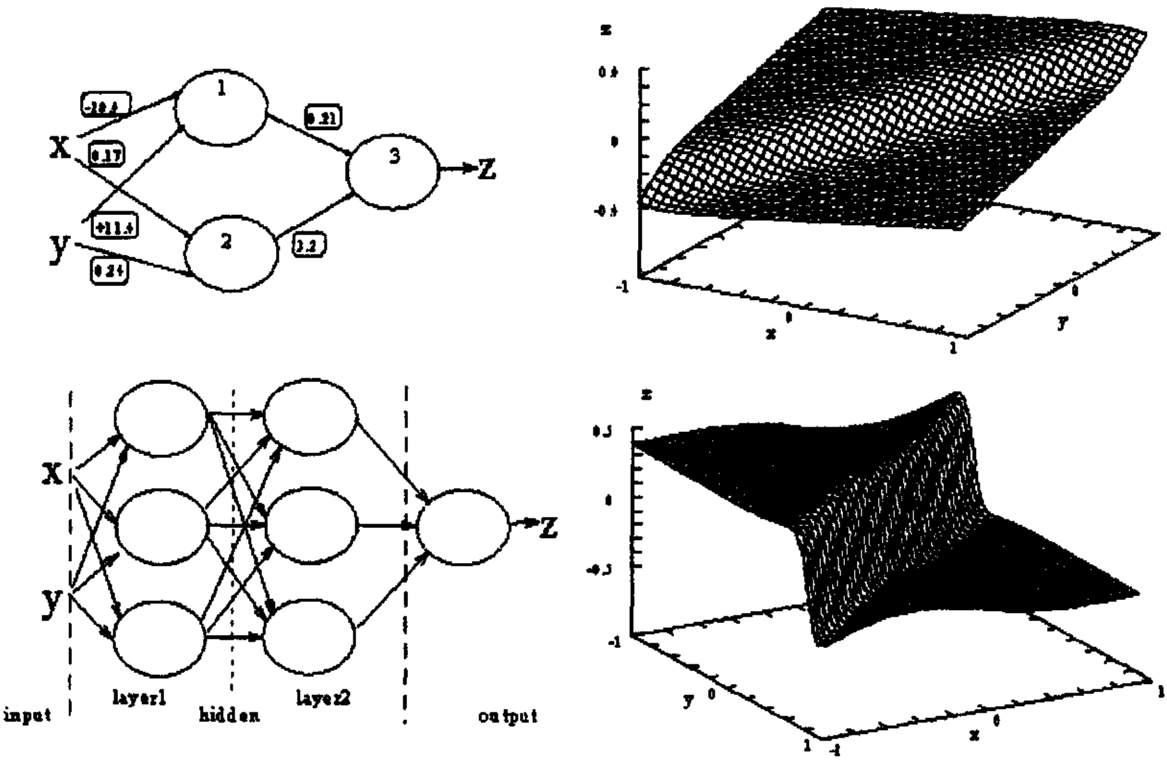

Figure 5-10: Neural networks consisting of several neurons.

Figure 5-10 shows typical neural networks: several connected neurons. In the top part, neuron 1 and neuron 2 are the top respectively the bottom of the examples of Fig.5-9. The

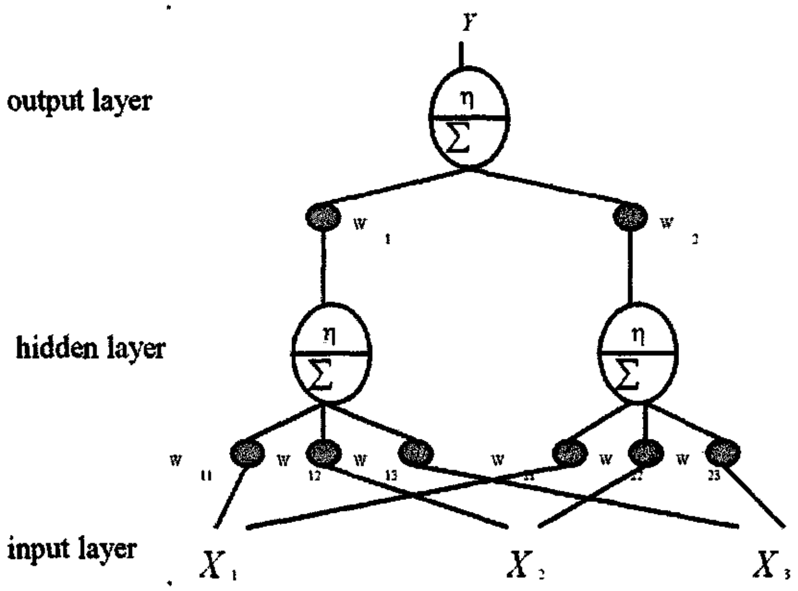

Figure 5-11: Parametrization of a neural newtork model.

outputs of these two neurons are input to a third neuron. The right part shows the surface generated by this neural network model. It is clearly the combination of the surfaces of Fig.5-9.

The bottom part of Fig.5-10 shows a more typical feedforward neural network. It has two inputs and one output. In between there are two "sandwiched" layers of hidden neurons. Each connection in the figure has its own weight (not shown). The bottom right part of Fig.5-10 shows that neural network with sufficient complexity can produce surfaces of an arbitrary form.

The user of a neural network has to specify the complexity of the network, expressed in number of hidden layers and the weights for each connection. These weights are the parameters of the neural network. An explicit parametrization of yet another neural network example (Fig.5-11) is given by formula :

$$y = \eta(w_1\eta(w_{11}x_1 + w_{12}x_2 + w_{13}x_3) + w_2\eta(w_{21}x_1 + w_{22}x_2 + w_{23}x_3)) \tag{5.22}$$

The weights $(w_1, w_2, w_{11}, ...)$ have to be calibrated by an optimization procedure. To determine these parameters neural networks are trained by showing them observed input and output examples. By optimizing the weights the network learns the desired relation between input and output. So the MLFNN can be thought of as a *global function approximator.* According to Kolmogorov's theorem (see Kurkova [34]) such networks can approximate arbitrarily well any function (one-one or many-one) from one finite-dimensional space to another, provided the number of hidden units is sufficiently large. The problem is, however, that Kolmogorov's theorem neither gives a prescription how many hidden units are needed to solve particular problem, nor specifies what kind of transfer functions should be used. Moreover, training the network is usually difficult multidimensional optimization task. Regardless of the optimization method used it can be never guaranteed to find a global minimum of the error function of the network.

Parzen generalized regression as a PNN

The technique described in Section 4.2 is also known under the name *Probabilistic Neural Networks*, or PNN for short. There are several reasons why they are called neural networks :

1. They generalize the probabilistic view on classical regression, just as feedforward neural networks generalize the algebraic view;
2. They have a massive parallel architecture, as all neural networks (artificial or not).
3. Their calculation and the corresponding optimization of the bandwidth can be made within the framework of a classical MLFNN, as Specht [54] has shown.

Figure 5-12 illustrates this concept. PNN is just a parallel implementation of Eq.(4.53) :

$$g(\mathbf{x}) = \mathrm{E}[\mathbf{Y}|\mathbf{X} = \mathbf{x}] = \frac{\sum_i f_{\mathbf{X}}^{(i)}(\mathbf{x})\ \mathbf{m}_{\mathbf{Y}|\mathbf{X}=\mathbf{x}}^{(i)}}{\sum_i f_{\mathbf{X}}^{(i)}(\mathbf{x})} = \frac{N(x)}{D(x)}$$

It consists of the input layer, the pattern layer, the summation layer and one output neuron. Each layer consists of a number of processing units (neurons). Connections between the layers, parametrized by certain coefficients (weights), indicate a signal flow within the network. Input layer has only a symbolic character and represents the place where the external input sequentially enters the network. The number of neurons in this layer equals the dimension of the input vector x, which in this case consists of three variable elements (x_1, x_2, x_3). Connections

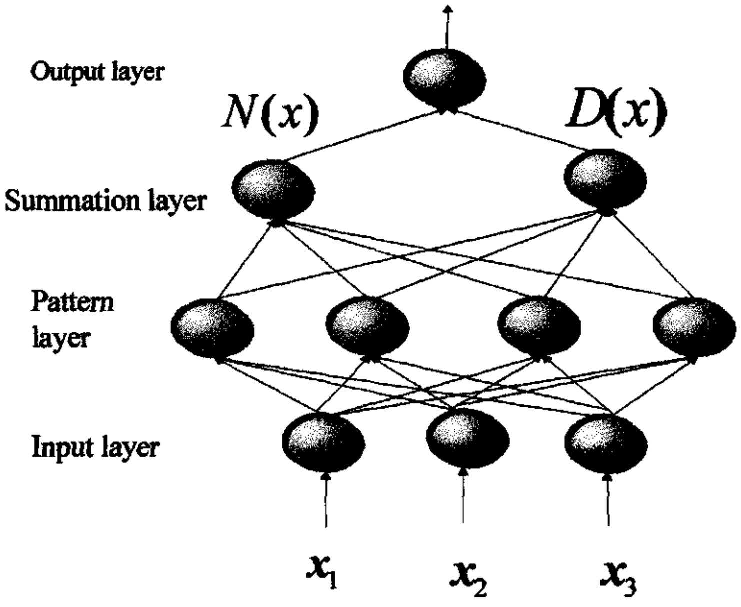

Figure 5-12: A Probabilistic Neural Network (adapted from Specht [54]).

between the input and the pattern layer have a unit value. The actual processing takes place in the pattern layer, which contains *one neuron per training case* $\mathbf{x}_i$ ($i = 0, 1, ..., 3$ here) Each of these neurons calculates $f_{\mathbf{X}}^{(i)}(\mathbf{x})$ that depends on *Mahalanobis distance* (see Eq.(4.14) in Section 4.1.3) between particular training case $\mathbf{x}_i$ and actual input $\mathbf{x}$ simultaneously presented to all pattern units.

Each pattern unit connects to both summation layer units. The left one represents the numerator $N(\mathbf{x})$, and the right one the denominator $D(\mathbf{x})$ of the fraction (4.53). This implies that weights of connections between the pattern layer and the left summation unit must take the values of $\mathbf{m}_{\mathbf{Y}|\mathbf{X}=\mathbf{x}}^{(i)}$. On the other hand, weights of connections between the pattern layer and the right summation neuron are of unit value. Once again looking back at the Eq. (4.53), we can see that the output neuron performs a simple division, which is the only non-neural (biological neurons are not capable to divide) operation in this architecture.

The training is straightforward: we minimize the least square criterion outlined in Section 4.2.4 by adjusting the σ value as shown in Fig.5-13 Thus, unlike in the case of the MLFNNs we optimize only one parameter. Besides, the PNN models can easily be made *local* by using local

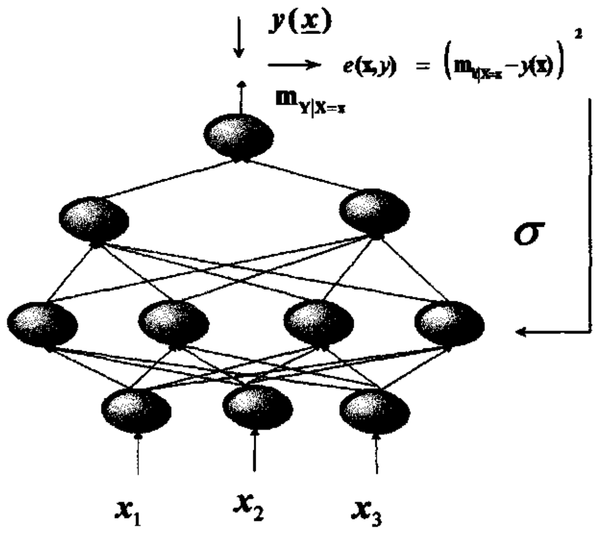
$y(\underline{x})$
$e(\mathbf{x},y) = \left(m_{Y|X=x} - y(\mathbf{x})\right)^2$
$m_{Y|X=x}$
σ
x_1
x_2
x_3

Figure 5-13: Training the PNN.

covariance matrix estimation. This direct localization scheme is not available for the MLFNN's - global approximators with, on the other hand, good localization properties.

Neural networks in hydroinformatics

In recent years an increasing trend has been observed to use neural networks for the modelling of input - output relations within science of hydroinformatics. The applications which appear in the hydrological literature most frequently are: rainfall - runoff modelling (see e.g. Hsu et al. [31], Dimopulos et al. [15], Lorrai et al. [37], Minns and Hall [41]) and forecasting of hydrological variables such as waterlevels or discharges (Akker and Viel [7], Boogard et al. [12], Wójcik [65]). Besides this, Lula and Pociask Karteczka [38] used neural networks for evaporation estimation in a small catchment. Moreover, Hall and Minns [30] used the MLFNN for estimating the quantiles and parameters of flood frequency distribution in ungauged catchment out of the catchment and rainfall characteristics. Another far interesting application was reported by Hanish et al. [29] where a neural net was adapted to predict wastewater treatment plant parameters.

Neural networks may also serve as an excellent tool for modelling and analysis of spatially distributed systems. They provide a good alternative for classical linear interpolation methods like krigging. A comparison between the two methods[3] in the context of elevation modelling for Walker Lake is outlined by Dowla and Rogers [18] p. 159-172. Generation of hydraulic conductivity fields was a subject of study carried out by Torfs and Bier [61]. Here a neural network was merged with a finite-element code to simulate the hydraulic conductivity patterns in order to obtain the best prediction of measured groundwater level. Dowla and Rogers [18] again (p.59-91) give an example of coupling genetic algorithm with neural network to work out an optimal strategy for ground water remediation.

The above view to the hydrological literature on neural networks is in no meaning complete and systematic. The intention of the author was just to inspire the reader by showing the broad spectrum of possible applications of this rapidly developing field.

[3] In this study Radial Basis Function network was used as an interpolator.

Part III

Analysis

Chapter 6

Case study and results

The aim of this study was to compare various black-box techniques for discharge prediction on the example of Hupselse-Beek catchment in the Netherlands. In this Chapter the results of this investigation will be presented. In the first Section some standard characteristics of the catchment in question are given. Afterwards, the data set used for making predictions is analyzed in terms of standard and non-linear statistics. Finally, the results of discharge forecasts by different models are shown and analyzed.

6.1 Hupselse Beek catchment description

Situation

The Hupselse Beek catchment covers about 6.5 km^2 in the east of the Netherlands near the village of Eibergen and the town of Grenlo (see Fig.6-1), its altitude varies between 24 and 33 meters above mean see level. The outlet of the Hupselse Beek catchment is 6° 38$'$E and 52° 04$'$N. The average slope of the area is 8 $^\circ/_{oo}$.

Water courses

The main stream is 4 km long. There are 7 small tributaries, varying in length from 300 to 1500 m. The average slope in the main stream is 1 $^\circ/_{oo}$. In the main stream there are 2 flumes and 5 overflow structures. This situation is presented in Fig.6-2

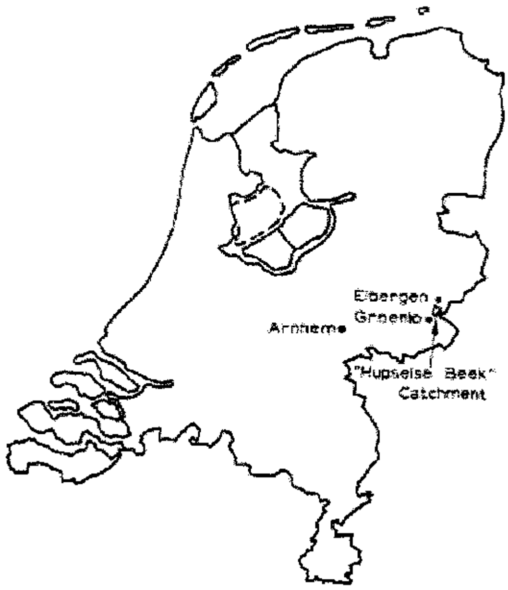

Figure 6-1: Hupselse Beek catchment in the Netherlands.

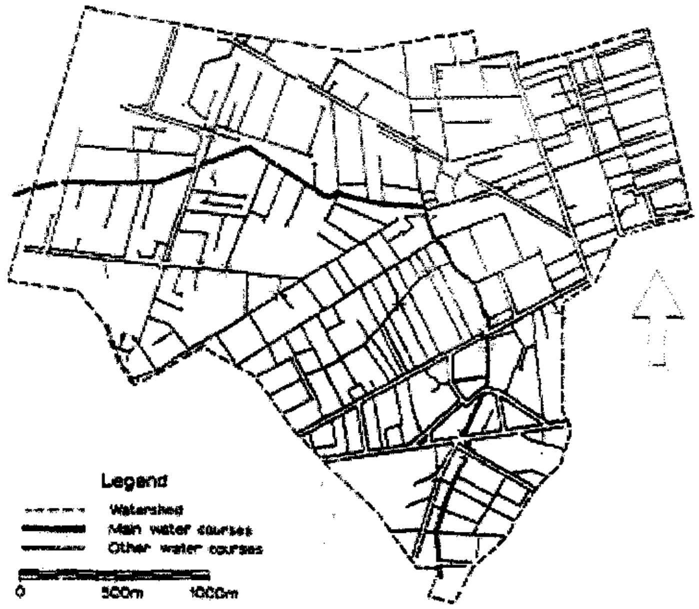

Figure 6-2: Map of water courses.

Variable\\Month	J	F	M	A	M	J	J	A	S	O	N	D	Year
precipitation (mm)	64.8	57.2	45.6	50.8	48.5	61.7	81.4	81.7	68.3	64.6	65.5	62	758.1
evaporation (mm)	4	15	39	73	106	120	112	92	58	26	9	3	657
temperature (C)	1.5	2.1	5.7	9.8	14.4	17.5	19.9	18.5	15.4	10.4	6.0	2.9	10.3
vapour pressure (mbar)	6.1	6.1	6.9	8.3	10.6	13.1	15.2	15.4	13.5	10.5	8.2	6.8	10.1
relative humidity (%)	85	82	75	69	66	67	71	74	78	82	86	88	77
sunshine hours	50	70	120	190	210	210	190	180	140	100	50	40	1550

Table 6.1: Average values over 30 years, Winterswijk

Geology and soil character

The Hupsel valley probably originated in the Middle Pleistocene, when the Rhine system deposited coarse sand and gravel in the area. Remnants of glacial till in varying thickness indicate that glacier tongues invaded this area during the Saalien. The present valley must have been shaped during the most recent glacial period. The most important deposits that can be found at or near the surface are: eolian, glacigenous, fluviatile and marine deposits. Two levels with low permeability occur. Miocene clay is found at or near the surface in the eastern part of the area. In western direction remnats of glacial till are found at about 1 meter deep.

From a pedological point of view the area primarily consists of hydropodzol. slightly loamy sand. Landuse is mainly agricultural: about 70% is covered with grass, 21 % is areable land, 6% woodland and 3% wasteland.

Climate

The Netherlands has a maritime climate due to the mainly western winds. Towards the east, however, the continental character increases. The yearly rainfall distribution and the occurrence of thunder storms are also influenced. Table 6.1 gives an overview of the averages of different meteorologic variables, calculated for station Winterswijk (about 10 km south-east of the Hupsel area). The averages have been calculated over a period of 30 years (1931-1960). Additionally, mean monthly precipitation and mean monthly runoff from the catchment over the period of 1969-76 are listed in Table 6.2

Variable\\Month	J	F	M	A	M	J	J	A	S	O	N	D	Total
precipitation (mm)	56.2	41.6	44.6	48.2	71.3	58.9	72.8	61.3	57.5	51.5	68.4	48.9	682.1
runoff (mm)	34.8	34.7	22.9	30.5	16.6	8.2	9.3	3.9	2.3	10.4	22.1	31.8	227.5

Table 6.2: Monthly precipitation and runoff totals of the Hupsel area

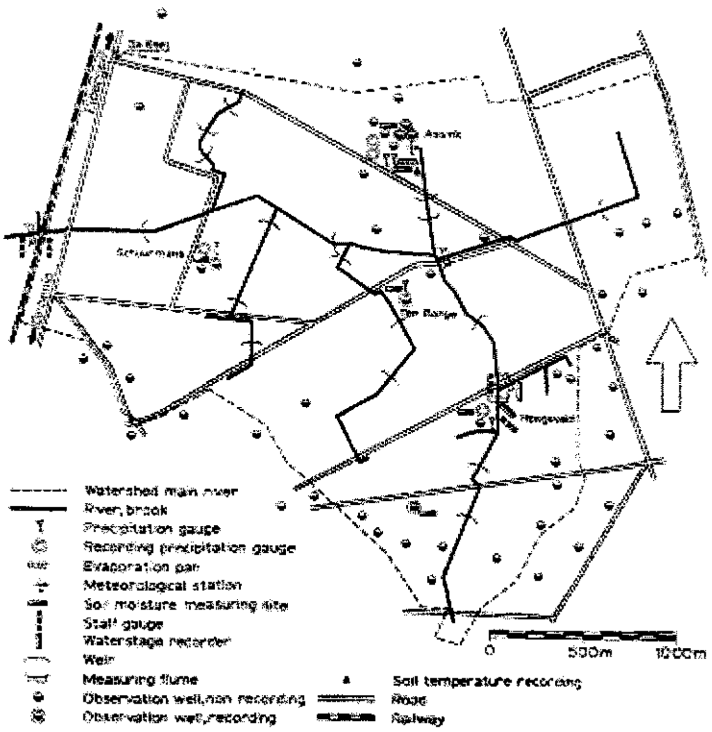

Figure 6-3: Hydrometeorologic network.

Measuring network

Figure 6-3 shows measuring equipment and hydrological network of the Hupsel area. The data set used in this project came from Assink meteorological station (rainfall information from rain gauge recorder) and from measuring fume (discharge information) situated on the main stream at the leftmost part of the picture.

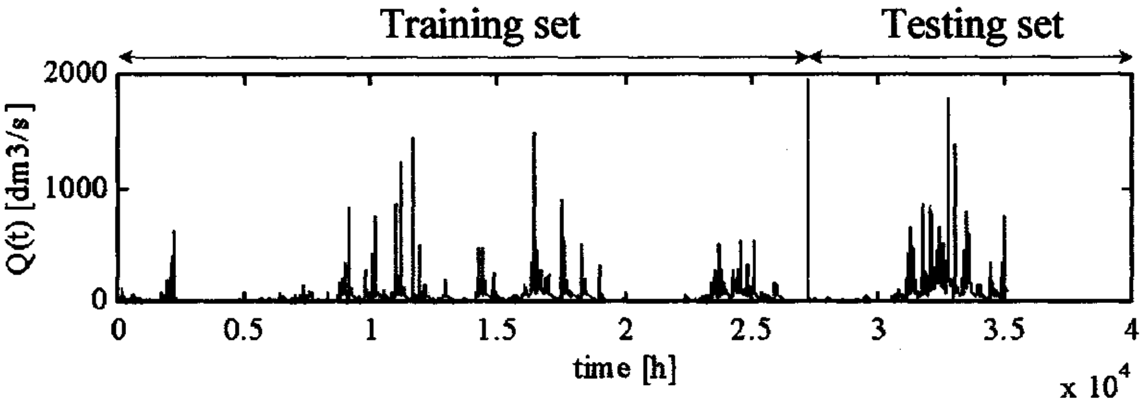

Figure 6-4: Discharge time series aggregated to hourly scale.

6.2 The data set

The data set used in this study all comes from Hupselse Beek watershed and contains time series of discharges $Q(t)$ [dm^3/s] aggregated to hourly scale from 15 min resolution data. This unique (with respect to the fine sampling scale) data collection was recorded between 01.04.1971-01.04.1975. The reason this example was chosen was because a lot of data on many different time scales were available. For the purpose of model making, the data set was broken into in two parts:

- **training set** for models' identification (01.04.1971 - 31.03.1974)

- **test set** for models' verification (01.04.1974 - 01.04.1975)

The time plot of the series is shown in Fig. 6-4 . Summary statistics are given in detail in Table 6.3. Histograms for total series, and separately for training and testing sets are presented in Fig.6-5. It is worth pointing out that the underlying marginal probability density is clearly non-Gaussian.

$^{Variable}\backslash_{Statistic}$	Nr of cases	Mean	Min value	Max value	Variance	Std	Skewness	Kurtosis
Q(t) training + testing set	35064	45.76	0	1793.75	7382.53	85.92	5.59	58.90
Q(t) training set	26304	38.33	0	1479.50	5091.06	71.35	6.49	73.85
Q(t) testing set	8760	68.04	0	1793.75	13602.53	116.63	4.13	30.69

Table 6.3: Summary statistics

6.3 Non-linear diagnostics

6.3.1 Nonlinearity detection in hydrological time series

In this Section a new idea which is aimed at detecting $\mathcal{LG}$ and non-$\mathcal{LG}$ non-linearities (defined in Section 2.2) in periodic or nearly periodic time series proposed by Stam et al.[55] will be described. As it will be argued further in this section, this method requires a simple modification to be specifically adapted to hydrological data mining. The objective of this analysis was to prove that the dynamics of the investigated time series cannot be characterized by simple $\mathcal{LG}$ model and requires special treatment in a form of non-linear models.

The basic concept

The main idea behind the method is that non-linearity can manifest itself in a (quasi) periodic time series in asymmetry in the amplitude distribution, time irreversibility or both. The two phenomena may occur independently, and should be characterized separately. According to Tong [60] *time reversibility* of a time series may be defined as follows: a stationary time series $\{X_t\}$ is time reversible if for every positive integer n , and every $t_1, t_2, ..., t_n$, the vectors $(X_{t_1}, X_{t_2}, ..., X_{t_n})$ and $(X_{-t_1}, X_{-t_2}, ..., X_{-t_n})$ have the same joint probability distributions. If this condition does not hold the time series is time irreversible. By analogy, the stationary, mean centralized time series is *amplitude symmetric* if the vectors $(X_{t_1}, X_{t_2}, ..., X_{t_n})$ and $(-X_{t_1}, -X_{t_2}, ..., -X_{t_n})$ are equal in joint probability distribution. The amplitude asymmetry and time irreversibility are indicators of $\mathcal{LG}$ and non-$\mathcal{LG}$ nonlinearity in the time series respectively. As it was mentioned already in Section 2.2 the former type corresponds to the situation when X_t is not observed directly, but undergoes some static transformation, either in the system itself, or in the measuring process. Formally, the time series $\{X_t\}$ can be transformed instantaneously to another time series $\{Y_t\}$ by a 1-1 function $h : Y_t = h(X_t)$ for each t. Now,

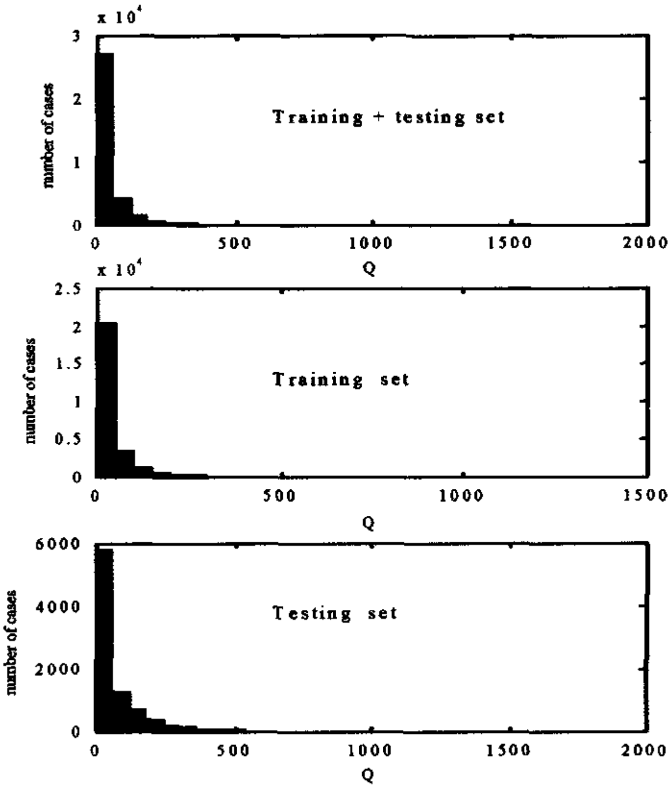

Figure 6-5: Estimated histograms for trainig and testing data sets.

$\{Y_t\}$ is stationary and time reversible if and only if $\{X_t\}$ is stationary and time reversible. In other words, the time irreversibility cannot be induced by the static transformation h. On the contrary, the amplitude asymmetry could, at least in principle, be explained by static nonlinear transformation of a stationary $\mathcal{LG}$ process, or in other words by non-Gaussianity of the probability density function (Palus [44]). Thus, if the observed time series $\{Y_t\}$ is time irreversible, the dynamics of the underlying $\{X_t\}$ cannot be explained by a stationary linear Gaussian process. That is why the nonlinearity associated with irreversibility concept was abbreviated as non-$\mathcal{LG}$. On the other hand, that this kind of nonlinearity does not necessarily imply that the time series $\{Y_t\}$ is generated by non-linear deterministic system. There are several reasons for this:

- the underlying non-linear dynamics may be generated by stochastic system (Eq.(2.4))

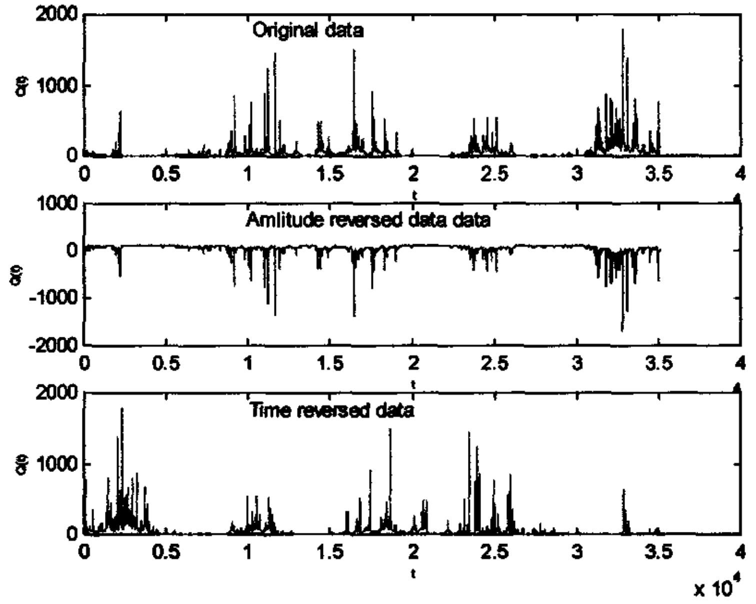

Figure 6-6: Detecting non-linearity: the time series used in the test.

- the measurement function may be not instantaneous $h : Y_t = h(X_t, X_{t-\tau})$

- the non-linear regime may occur locally due to violation of stationarity assumption on data

- filtering, especially differencing time series may lead to spurious results

On the other hand, the time reversible and amplitude symmetric signal does not imply that the underlying system must be linear. This situation might happen when the non-linear system produces a periodic solution. A good example given in Stam et al. [55] is the Poincare oscillator. The control parameters of this non-linear system can be adjusted in a way to generate a sine wave that is amplitude symmetric and time reversible.

Method description

The method consists of two elements:

- any nonlinear prediction model (see, e.g., Section 5.2.1)

- Amplitude and time reversed versions of data (Fig.6-6)

The amplitude reversed version is obtained as follows:

$$Z_t^{AR} = 2\langle X_t \rangle - X_t \tag{6.1}$$

where $\langle X_t \rangle$ is the mean of the original data X_t. The time reversed version is expressed by:

$$Z_t^{TR} = X_{N-t+1} \tag{6.2}$$

for $t = 1, 2, ..., N$ where N is number of samples in the time series. In the next step the nonlinear model of the dynamics *calibrated upon the original time series* is used to predict: (A) the original time series, (B) the amplitude reversed version, and (C) time reversed version. The procedure is designated as *nonlinear cross-prediction* since the time series used for calibrating the model and the time series that has to be predicted are different in case (B) and (C). After making the predictions, a plot of the correlation coefficient between predicted and actual time series is obtained for (A)-(C) as a function of prediction horizon. From now on this plot will be referred to as *predictability plot* (see Fig.6-8) Regardless on whatever the shape of (A) is, for a time series that is asymmetric around its mean value, predictability of (B) will be less than for (A). The same is true in the case of time irreversible time series : predictability for (C) will be less than for (A). Moreover, for stationary and reversible time series predictability for (C) will equal the predictability of (A). This can be seen by comparing the predictability plots for (A) and (C).

As suggested by Stam at al. [55] the simplified quantitative measure of how strong time irreversibility and amplitude asymmetry effects are, can be obtained by averaging the correlation coefficients over prediction horizons (in this study 24 horizons were used). This averaged correlation coefficient is designated "pred" for (A). The amplitude asymmetry (ama) is defined as a difference between the average correlation coefficient for (A) and (B). Similarly, the time irreversibility is defined as a difference between the average correlation coefficient for (A) and for (C). The Amplitude asymmetry in the time series will result in ama>0; time irreversibility in tir >0. Consequently a key to the nonlinearity detection can be outlined as follows:

- if ama $\preceq 0$ and tir $\preceq 0$ one cannot reject the hypothesis whether the underlying system

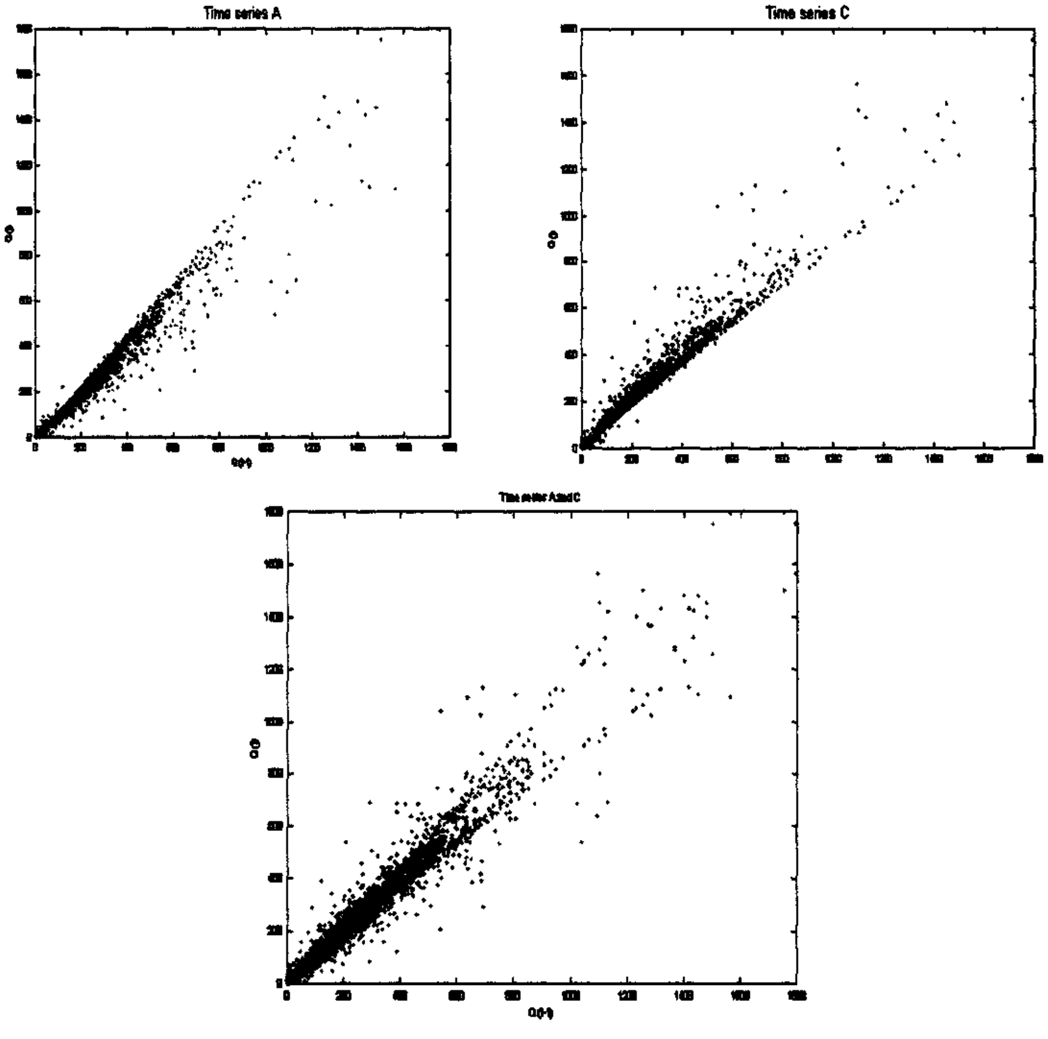

Figure 6-7: Time delayed scatterplots of original and time reversed discharge data.

Discharge range	Count	Cumul. Count	% of valid	cumul % of valid
$Q = 0$	1714	1714	4.88	4.88
$0 < Q \leq 100$	29045	30795	82.83	87.72
$100 < Q \leq 200$	2758	33517	7.86	95.58
$200 < Q \leq 300$	844	34361	2.40	97.99
$300 < Q \leq 400$	327	34688	0.93	98.92
$400 < Q \leq 500$	171	34859	0.49	99.42
$500 < Q \leq 600$	85	34944	0.24	99.65
$600 < Q \leq 700$	47	34991	0.13	99.79
$700 < Q \leq 800$	27	35018	0.08	99.87
$800 < Q \leq 900$	14	35032	0.03	99.91
$900 < Q \leq 1000$	6	35038	0.02	99.92
$1000 < Q \leq 1100$	5	35043	0.01	99.94
$1100 < Q \leq 1200$	5	35048	0.01	99.95
$1200 < Q \leq 1300$	5	35053	0.01	99.96
$1300 < Q \leq 1400$	2	35055	0.01	99.97
$1400 < Q \leq 1500$	5	35060	0.01	99.98
$1500 < Q \leq 1600$	2	35062	0.01	99.99
$1700 < Q \leq 1800$	0	35062	0.00	99.99
$1800 < Q \leq 1900$	2	35064	0.01	100

Table 6.4: Empirical frequency table for Q(t) series

is linear or non-linear

- if ama > 0 and tir $\preceq 0$ there is an indication of $\mathcal{LG}$ nonlinearity; there is not enough evidence to indicate non-$\mathcal{LG}$ nonlinearity in data
- if ama > 0 and tir > 0 there is a sign of the non-$\mathcal{LG}$ nonlinearity of data
- if ama $\preceq 0$ and tir > 0 there is a sign of the non-$\mathcal{LG}$ nonlinearity of data

Hydrological modification

The concepts of amplitude asymmetry and time irreversibility, by definition, are dependent on a criterion according to which one compares joint probability distributions. The problem with hydrological time series is that asymmetry in distributions of actual time series and time-reversed time series occurs locally, only for the extreme events. This effect is exemplified on Figure 6-7 where the total data collection of discharges $Q(t)$[1] used in this study (training

+testing set), is plotted against the 1h delayed discharges $Q(t-1)$ for both (A) and (C) time series (vide Figure 6-6). In the upper two pictures (A) and (C) series are plotted separately. The question now is whether two 2-D probability distributions of points in the two plots (one may think of these distributions as of an ink density on the two pictures) would be sufficiently different to indicate time irreversibility. The answer is negative because the majority (87%) of the data (see Table 6.4) lies within $\langle 0; 100)$ interval. The points are densely concentrated there forming a peak in the 2-D distribution. If one overlaps the two plots (see lower panel of Figure 6-7), it may clearly be seen that for $Q(t) \epsilon \langle 0; 100)$ the patterns match each other almost perfectly, i.e., they are symmetric. However, this symmetry only holds for the specified interval. And here comes the "hydrological" modification of the nonlinearity test; instead of calculating the correlation coefficients for the predictability plots on the basis of the whole range of the predicted and the actual time series, one may impose a certain threshold for this calculation. So, to once again clarify the procedure, the following steps are to be taken :

1. calibrate nonlinear model using (A) series
2. make cross-predictions of (C) series
3. select those elements of the *original* (C) series that > threshold^C
4. select the *corresponding predicted* (C) series
5. for each prediction horizon calculate correlation coefficient between the 3) and 4) series
6. make predictability plot
7. calculate tir for the imposed threshold
8. vary the threshold and go to 2)

The above algorithm can be also adopted for (B) series, however, as it will be shown in the next section, amplitude asymmetry is evident for the discharge time series. The major change regards to 3) :

[1] It was not neccecary to make a distinction between training and testing set in this case since the aim of this exercise was to quantify various forms of nonlinearities, not the quality of the prediction models.

Threshold	pred	ama	tir
No	0.89	0.29	0.01
150	0.73	0.69	-0.03
200	0.68	0.67	0.00
250	0.64	0.64	0.04
300	0.62	0.61	0.14
350	0.66	0.64	0.18
400	0.67	0.66	0.16

Table 6.5: Pred, ama and tir coefficients for different thresholds for predicted and actual time series averaged over 1-24 prediction leads. In all cases the original time series is used to predict (A) the original time series, (B) amplitude reversed time series and (C) time reversed time series.

- select those elements of the *original* (B) series that are smaller than threshold^B, where:

$$threshold^B = 2\langle Q(t)\rangle - threshold^C \tag{6.3}$$

Results with discharge series

Non linear cross-prediction was applied to all (A),(B),(C) data. The state space local linear model with $d_L = 4$ and $T = 1$ was used. For each time series, and for each threshold pred, ama and tir based on averaging over 1-24 prediction horizons were calculated. The results are shown in Table 6.5. When no threshold was imposed, the data show the strong evidence of amplitude asymmetry (ama=0.29) and no sign of dynamic nonlinearity. Referring to the previous section, the rule of the majority hides the important details of the discharge dynamics. Whereas, the tir coefficient is marginal low for thresholds 150-250 , the tir for 300-400 thresholds detects non- $\mathcal{LG}$ nonlinearity in the time series. This illustrates very clearly that the results are dependent upon the criterion (in this case "thresholded" predictability plots) one adopts to compare joint probability distributions. Figure 6-8 once again illustrates the importance of the criterion choice. In the upper picture, showing the results of the original version of the method, the predictability of (A) series coincides with the predictability of (C) series. The predictability of the amplitude reversed data (B) is substantially worse, and independent of the prediction horizon. On the other hand, when modified version of the method was used,

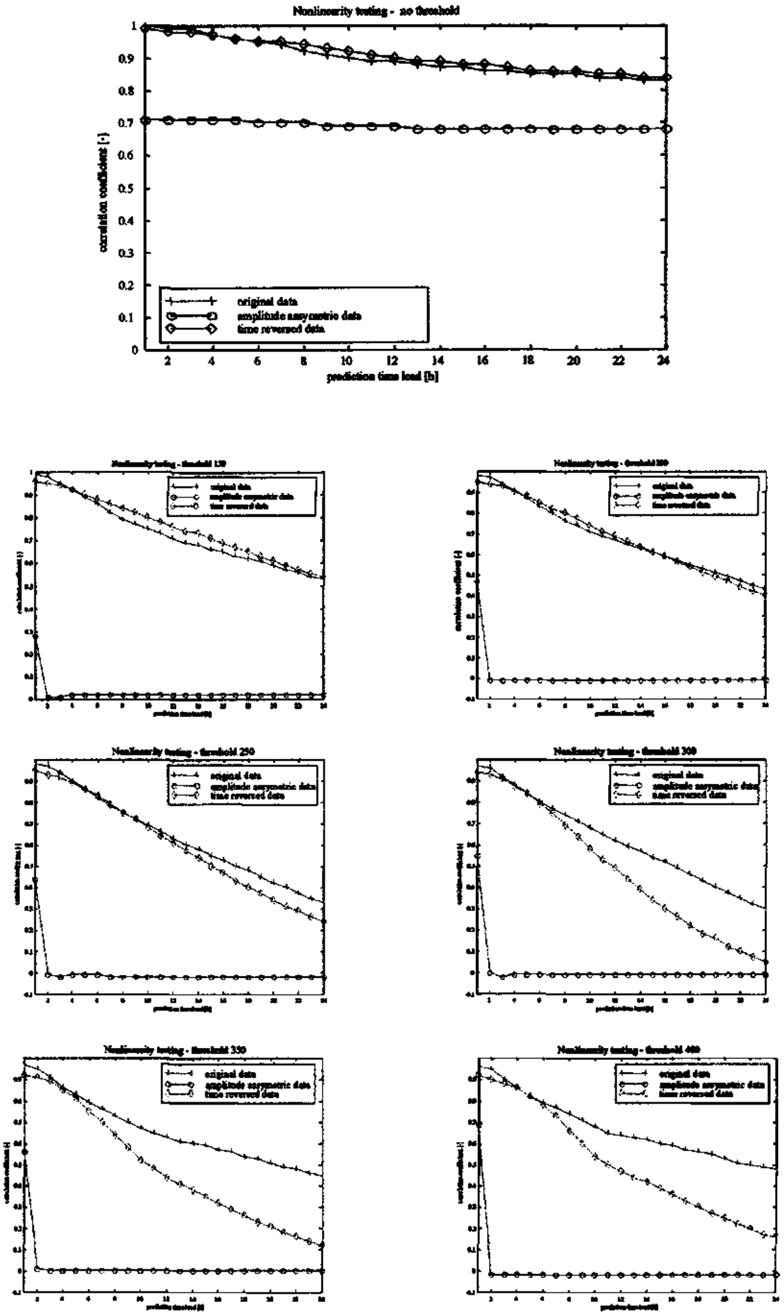

Figure 6-8: Predictability plots for discharge time series (A),(B) and (C). Original and modified version of nonlinearity detection method.

the results were dependent of the threshold imposed. In all cases (threshold 150-400) the amplitude asymmetry was present. The time irreversibility starts to be apparent already at threshold of 250, for which the predictability of (C) series gets worse than the predictability of (A) series as prediction horizon increases. It is important to notice , however, that, because of averaging over prediction horizons, this information is no clearly indicated in Table 6.5. The systematic decrease in predictability of (C) data as compared to (A) time series strenghtens for higher threshold values, showing local[2] effects of non- $\mathcal{LG}$ nonlinearity. In geophysics this phenomenon is known as *intermittency*. Simply, there are regions in the time series where non-$\mathcal{LG}$ nonlinearity exists, and there are regions as well where the time series exhibits linear or statically nonlinear behavior . In the discharge time series, the two situations might correspond to:

1. Rapid hydrograph spike formation due to new rainfall (a highly nonlinear phenomena as, e.g., Georgakos et al.[23] pointed out).
2. Decay of discharge after rainfall and baseflow dynamics which mostly depends on groundwater fluctuations (a linear - reservoir phenomena).

Discussion and cautionary remarks

Using the method described above we were able to indicate the presence of two kinds of nonlinearities in the discharge data:

- globally, the $\mathcal{LG}$ non-linearity
- locally , the non $\mathcal{LG}$ non-linearity

The former effectively means that the underlying distribution of the data is non-Gaussian. Recalling the discussion from Section 2.3 the interesting question is whether this non-linearity should be tackled by trying to inverse the static transformation (at least in algebraic sense using e.g. Box-Cox mapping) and then fitting a linear model to data or to model the data "as such" by non-linear methods. There are some general arguments that the second approach is more appropriate and easier to apply:

[2] For extreme discharge events.

1. By Gaussianizing the data we can only assure that the marginal density will be Gaussian. There is no proof, however, that multivariate densities (at least up to the dimension of the time series model input) will be Gaussian as well. One could test for Gaussianity of the joint densities, but those procedures especially in higher dimensions are numerically very difficult.

2. As indicated in Table 6.4 4% of the data contains 0 discharges. These events should be treated as "atoms" or namely points having discrete probabilities of occurrence. The correct approach would be to represent the total distribution as a superposition of "atoms" and remaining continuous part. Additionally, it is not possible to Gaussianize atoms using 1-1 transformation (parametric or not).

3. The analysis of dimensionality of the data (see Section 6.3.3) shown that while the global embedding dimension is 6 the local dynamics takes place in sub-space of dimension 4. This is never the case with $\mathcal{LG}$ processes that tend to spread the data filling total of space. Moreover, the locality of the dynamics cannot be reversed by 1-1 transformation. The important properties of the underlying process would simply be lost.

The dynamic nonlinearity detected by the thresholded version of the method is much stronger argument for the use of non-linear models, however some caution should be taken to interpret this results. In this study we did not test for stationarity in our data. It could turn out that the discharge series in considered period of time were non-stationary .This could introduce some artifacts to all calculations. What is more, as indicated in Section 6.2 we aggregated 15 min discharges to hourly scale. This procedure could have some impact on the results as well.

On the other hand, the non-linear models we investigated in this study can in principle tackle all the above mentioned difficulties much better then liner models. The last statement especially refers to the class of local non-linear models.

6.3.2 Mutual information - revisited

The data of discharge time series from Hupselse Beek was used to calculate $I(T)$ and $IL(T)$ functions introduced in Section 4.70. As can be seen in Fig. 6-9 shapes of the two functions are similar. $I(T)$ appears to be shifted version of $IL(T)$. According to Palus [44] this shift can

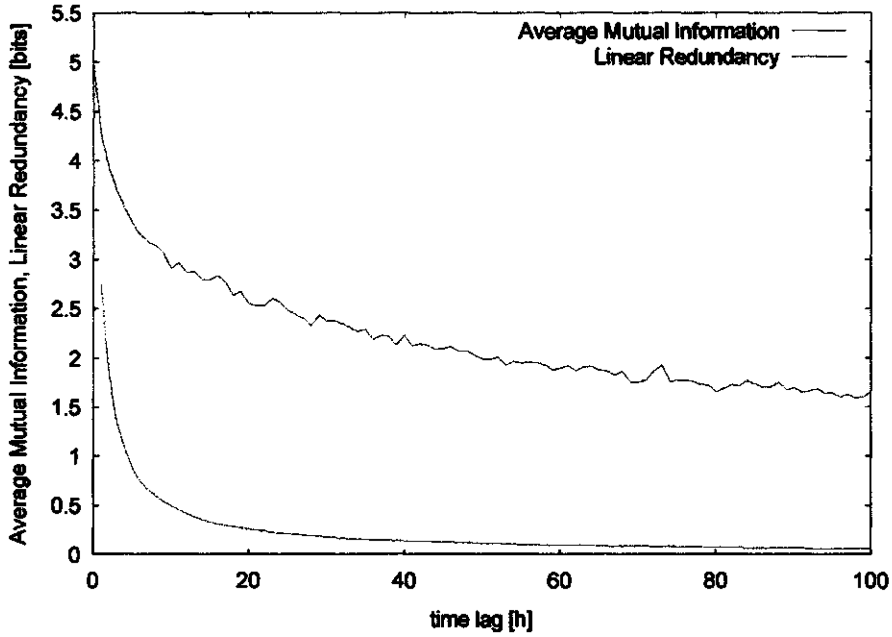

Figure 6-9: Mutual information and linear redundancy for Hupselse Beek discharge time series.

be explained by $\mathcal{LG}$ nonlinearity, which remains with agreement to the ("classical" : obtained without thresholding) results from the previous section. Non-linearity detection based on $I(T)$ allows, however, only global view to data .

Recalling considerations on choosing optimal time delay for a state space reconstruction by $I(T)$, we clearly see the function possesses no identifiable local minima. Fluctuations on $I(T)$ plot result from Monte Carlo integration procedure which is best shown in Fig.6-10 where different estimates of $I(T)$ are displayed. Generally, there is a marked increase in quality of $I(T)$ estimates with increase of number of Monte Carlo samples n (see Eq.(4.76)). The estimates are getting smoother and as $n \to \infty$ all the fluctuations will disappear.

6.3.3 Assessing dimensionality of discharge dynamics

With the help of methodology from Sections 5.1.4 and 5.1.5 embedding dimension and local dynamic dimension were calculated for discharge time series. This time only calibration part of the data was used since outcome from this analysis will further be utilized for identification

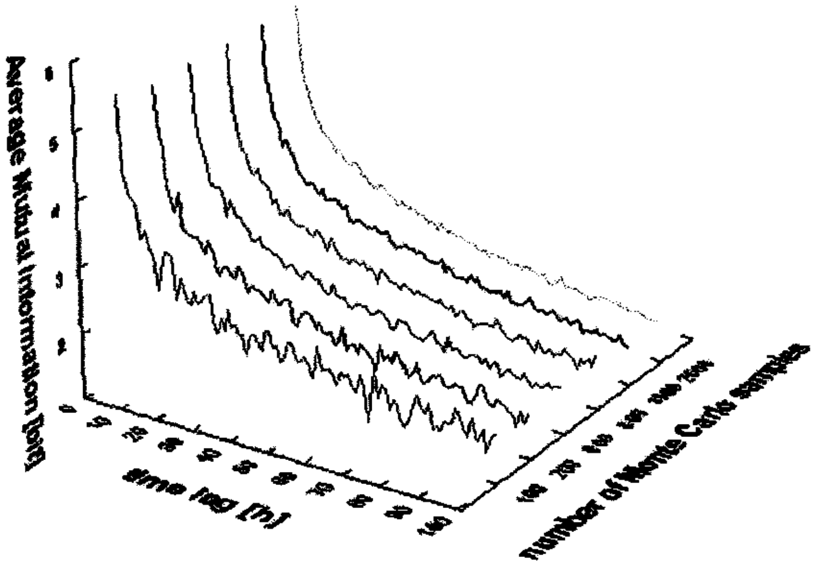

Figure 6-10: Mutual information by Monte Carlo integration.

of non-linear prediction models.

Embedding dimension

Using time lag $T = 1$ hour, the percentage of false nearest neighbors for discharge data shows a drop close to zero at $d_E = 6$ after which the (joint) percentage of false neighbors remains approximately constant (see Fig.6-11). This provides evidence that we are dealing with a low dimensional system. $d_E = 6$ indicates that in order to make discharge predictions by means of global non-linear models, as an input variables we have to use present discharge and 5 past (delayed) discharges (see Section 6.4).

It is interesting to notice that at dimension one percentage of FNN's is about 60%. Kennel et al.[33] found that if the data is clean from noise the percentage of FNN's at dimension one should be 100%. However, in the case of discharge series one should also take into account another explanation: especially low discharge periods are sometimes very persistent, what results in a lot of self-repeating points in the data set. These points remain always "true neighbors" to themselves.

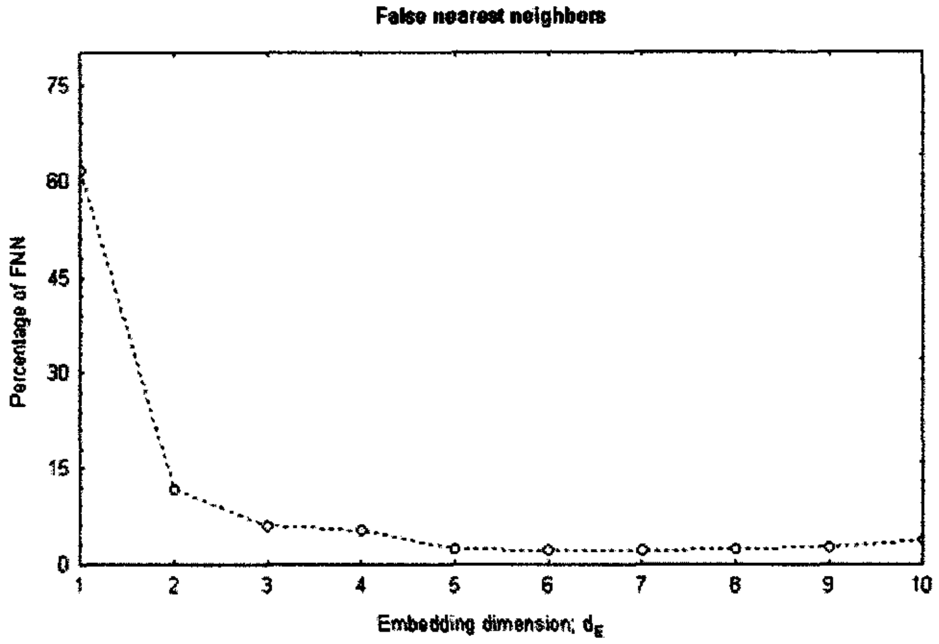

Figure 6-11: Embedding dimension analysis for the Hupselse Beek discharge series.

Additionally, Fig.6-11 reveals some deterministic structure present in data. Theoretically it is known (Abarbanel[1]) that for purely random signals percentage of FNN never saturates at near-zero level and determination of proper d_E will degrade.

Choosing the dynamical dimension

For the Hupselse Beek discharge data the percentage of bad predictions seen in Fig.6-12 becomes independent of the number of neighbors N_B and of the working dimension d at $d_L = 4$, telling us that *locally* the data points from reconstructed $d_E = 6$ state space occupy four dimensional subspaces. Therefore, to be able to make good predictions of future values of discharge time series with the help of *local* non-linear models as an input variables we have to use present discharge and 3 past discharges (see next Section).

6.4 Forecasting the discharge

Two classes of autoregressive models were used for discharge prediction in Hupselse-Beek:

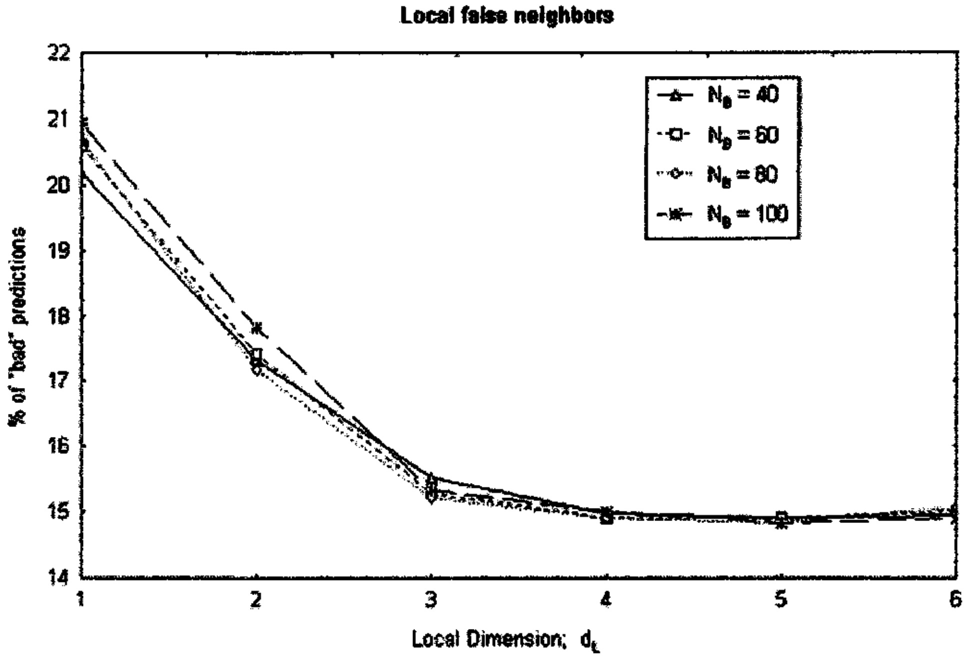

Figure 6-12: Local false nearest neighbours for discharges from Hupselse Beek. From this viev $d_L = 4$ may clearly be chosen.

1. Global models :

   (a) Linear AR

   (b) MLFNN 6-2-2-1 architecture[3]

   (c) Standard PNN

2. Local models:

   (a) PNN with locally estimated covariances (LPNN)

   (b) State space: P(1) as first and P(2) as second order polynomial (see Eq.5.18)

The forecasts with horizon ξ varying from 1 up to 12 hours ahead were based on present discharge and a few delayed discharges. The number of inputs (order) for AR model, estimated

[3] 6 inputs, 2 hidden layers consisting of 2 neurons each one, and 1 output neuron

according to Akaike Information Index (see Priestley [48] p.373) was 7. As suggested earlier, for global and local non-linear models the number of inputs was set to $d_E = 6$ and $d_L = 4$ respectively. Using the convention from Section 4.2 we make the following choices of inputs and outputs:

- for linear AR(7) model

$$\begin{aligned} \mathbf{x}_n &= (Q(n), Q(n-1), ..., Q(n-6))^T \\ \mathbf{y}_n &= (Q(n+\xi)) \end{aligned} \tag{6.4}$$

- for global non-linear models:

$$\begin{aligned} \mathbf{x}_n &= (Q(n), Q(n-1), ..., Q(n-d_E-1))^T \\ \mathbf{y}_n &= (Q(n+\xi)) \end{aligned} \tag{6.5}$$

- for local non-linear models:

$$\begin{aligned} \mathbf{x}_n &= (Q(n), Q(n-1), ..., Q(n-d_L-1))^T \\ \mathbf{y}_n &= (Q(n+\xi)) \end{aligned} \tag{6.6}$$

A measure of prediction accuracy was given by the normalized mean squared error:

$$NMSE = \frac{1}{VAR(\mathbf{y})N} \sum_{i=1}^{N} (\mathbf{y}_i - \widehat{\mathbf{y}}_i)^2 \tag{6.7}$$

where $\mathbf{y}_i$ was the value of i-th point of the testing set of length N, $\widehat{\mathbf{y}}_i$ was the predicted value, and $VAR(\mathbf{y})$ was the variance of the testing set. In other words $NMSE$ is the ratio of mean squared errors of the prediction method in question and the method which predicts the mean at every step. $NMSE = 1$ implies that the prediction method is equally good as taking the overall average of the time series.

6.4.1 Identification

Only the data points from the training set were used for constructing the models. While calibration of AR(7), P(1) and P(2) models was straightforward, some problems were encountered during training the neural nets. In the case of MLFNN the backropagation combined with Levenberg - Marquardt optimization method was used. For each prediction horizon, training procedure was repeated 10 times with epoch size set to 1000. Due to random selection of weights at the start , each training run resulted in different value of NMSE. Figure 6-13 shows the training results for 10 hours ahead prediction. The fluctuations in NMSE are obviously caused by local minima in error function of the network. It appeared that one initial set of weights gave better results than the other. To find the global minimum of the error function the learning procedure should be repeated for several times. Even then, however, there is no guarantee of not being trapped in local valley of the error function. From the practical point of view there is always a trade-off between calculation time and the quality of the results. In this exercise the networks was re-run only a few times (5h CPU time each net) and no claims about global optimality of those results are made.

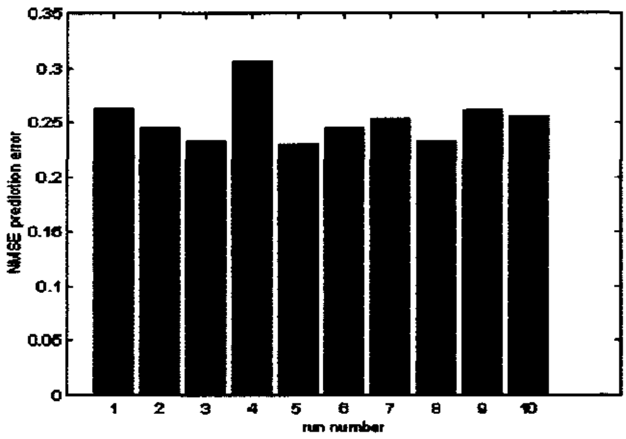

Figure 6-13: Training the MLFNN with Levenberg - Marquandt method.

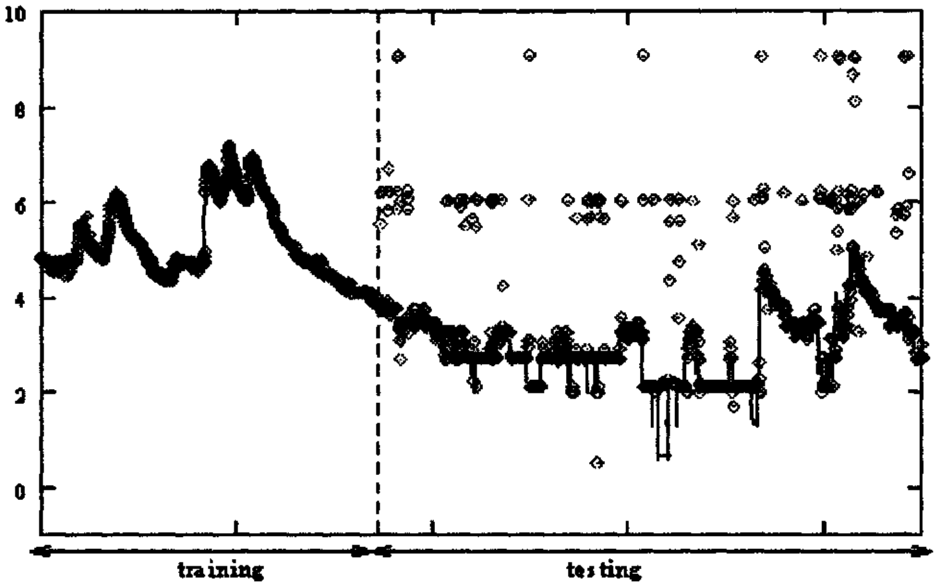

Figure 6-14: An example of overfitting.

As an opposite to the MLFNN, the PNN's training involved optimization of only one hyperparameter σ. For this purpose simple Brent technique (see Press et al. [47] p. 299-302) was used. While training went smoothly for the standard PNN, some experimentation showed that the LPNN suufered from the overfitting. The network perfectly reproduced the training set and was not able to generalize (to give good predictions) to the testing set. This phenomenon is best shown in Fig. 6-14. Predicted (blue dots) and measured (black line) discharges match each other only for the training part of data. The bursts of the predicted discharges in testing part indicate that the local PNN is unable to generalize the information learned from the training set. There are several strategies to tackle this problem :

- Split the training set into two parts : calibration and validation part. Construct the local Parzen densities using the calibration set and optimize the prediction of the validation set.

- Estimate the underlying probability density of the training set. Draw points randomly out of this density, center the Parzen kernels on these points and train the LPNN to predict the training set.

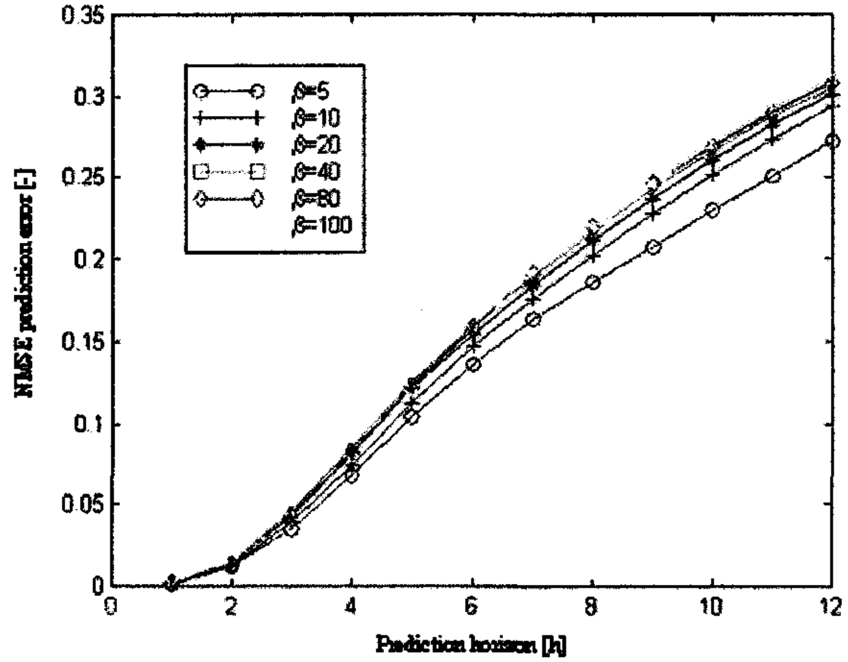

Figure 6-15: Sensitivity of the LPNN prediction on β choice.

- Set σ=1 without optimization and use the structure found by local covariance matrices in the training set to predict the testing set

In this study only the last approach was used. In neural network terminology the LPNN became a memory - based method. The network stored the training set in the memory and used this information straightforwardly to produce forecasts.

The last remark concerns the choice of β parameter, responsible for the locality of covariance matrices. Figure 6-15 shows the results of the training set prediction using different values of β. The lower the predictability line lies, the better the prediction is. Clearly, the smallest $\beta = 5$ yielded the best results. This value was used during verification of the LPNN.

6.4.2 Discussion of verification results

The models' performance was verified using an independent testing set. Since in this thesis the main attention was paid to the state space models and to the LPNNs, all the graphical impressions outlined here regard to only those two techniques. Figures 6-16 - 6-27 show the predictions 1,4,8,12 h ahead. The pictures represent the results for typical low, medium and high discharge regimes, selected from the testing set. For the P(1) model the prediction (blue dotted line) is plotted against measured discharge (green solid line). In addition residuals (red dotted line) are displayed, since there are no error bounds available for these forecasts. In contrast, for the LPNN, measured and predicted discharge is plotted together with error bounds calculated by the standard deviation.

Figures 6-16, 6-17 and 6-22,6-23 show the results for typical low discharge period. Characteristic for the Hupselse Beek is that periods of very low discharge are abundant. The discharge then is so low that the resolution of the measurements becomes visible. This creates extra problems for non-linear regression techniques. It is clear that P(1) and LPNN were able to overcome these problems. Figures 6-18, 6-19 and 6-24,6-25 present the results with the medium range discharges. Due to the locality of both algorithms the results are equally well in this regime. The same is true for Fig. 6-20, 6-21 and 6-26,6-27 where the high discharge was predicted. The striking discrepancy between P(1) and the LPNN is that when prediction horizon increases, the forecasts made by P(1) are much smoother than the forecasts made by the LPNN. The latter model exhibits some kind of peaky behavior which has to do with the different concept of locality incorporated into it. On the other hand, the artificial (or less probable) nature of the peaks is *always* indicated by broader standard deviation bands. This extra information provides a potential user with the insight into reliability of the predictions made by this non-conservative algorithm.

The quantitative comparison between the above two methods and the other forecasting techniques is presented by means of predictability plots (Fig. 6-28). The upper graph illustrates very clearly that except from the standard PNN, all non-linear models outperformed (on the average) the linear AR(7) model. This indicates presence of non-linear information in the studied data set and confirms the results obtained in Section 6.3.1. The best predictions were made by P(1) and P(2) models (the two lowest predictability curves). As pointed out by Casdagli [14] for complicated systems with large number of data points there are no advantages in using local quadratic predictors over linear predictors. Not surprisingly, the predictions using P(1) were the same (excluding small discrepancy at $\zeta = 9$) as those using P(2) model. The

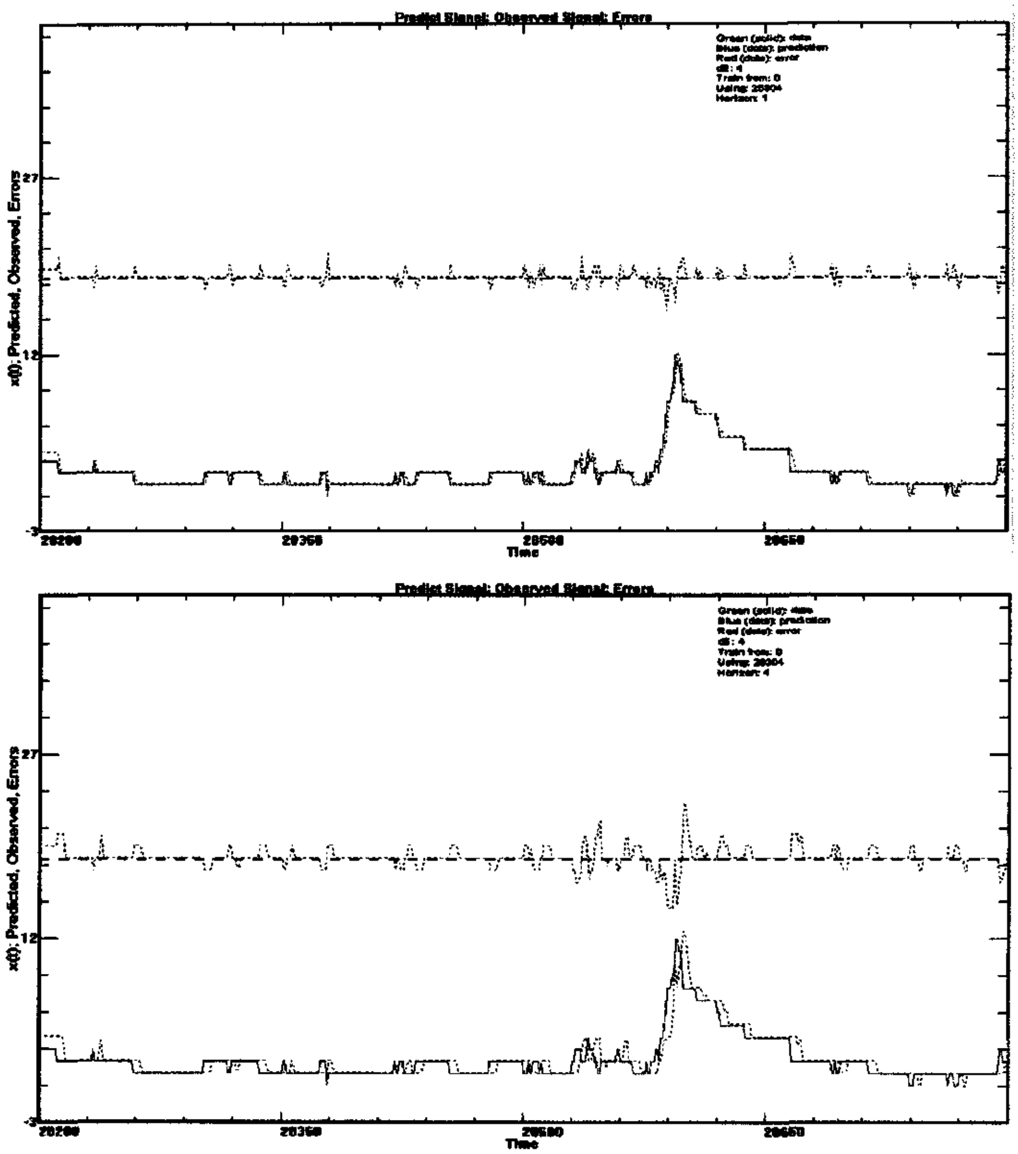

Figure 6-16: Discharge prediction by P(1) model. Low discharge period. Prediction horizon 1h (upper panel) and 4h (lower panel).

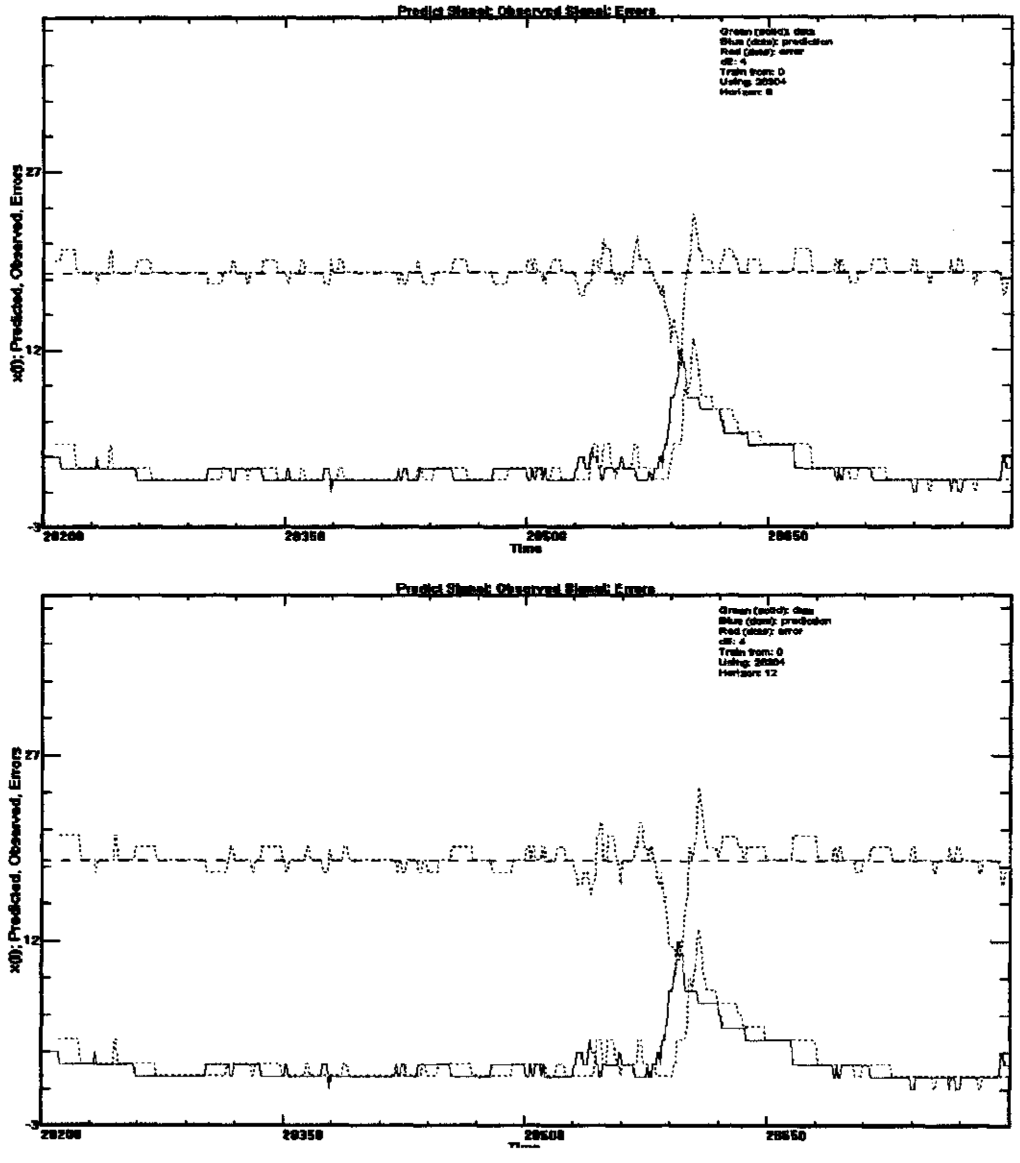

Figure 6-17: Discharge prediction by P(1) model. Low discharge period. Prediction horizon 8h (upper panel) and 12h (lower panel).

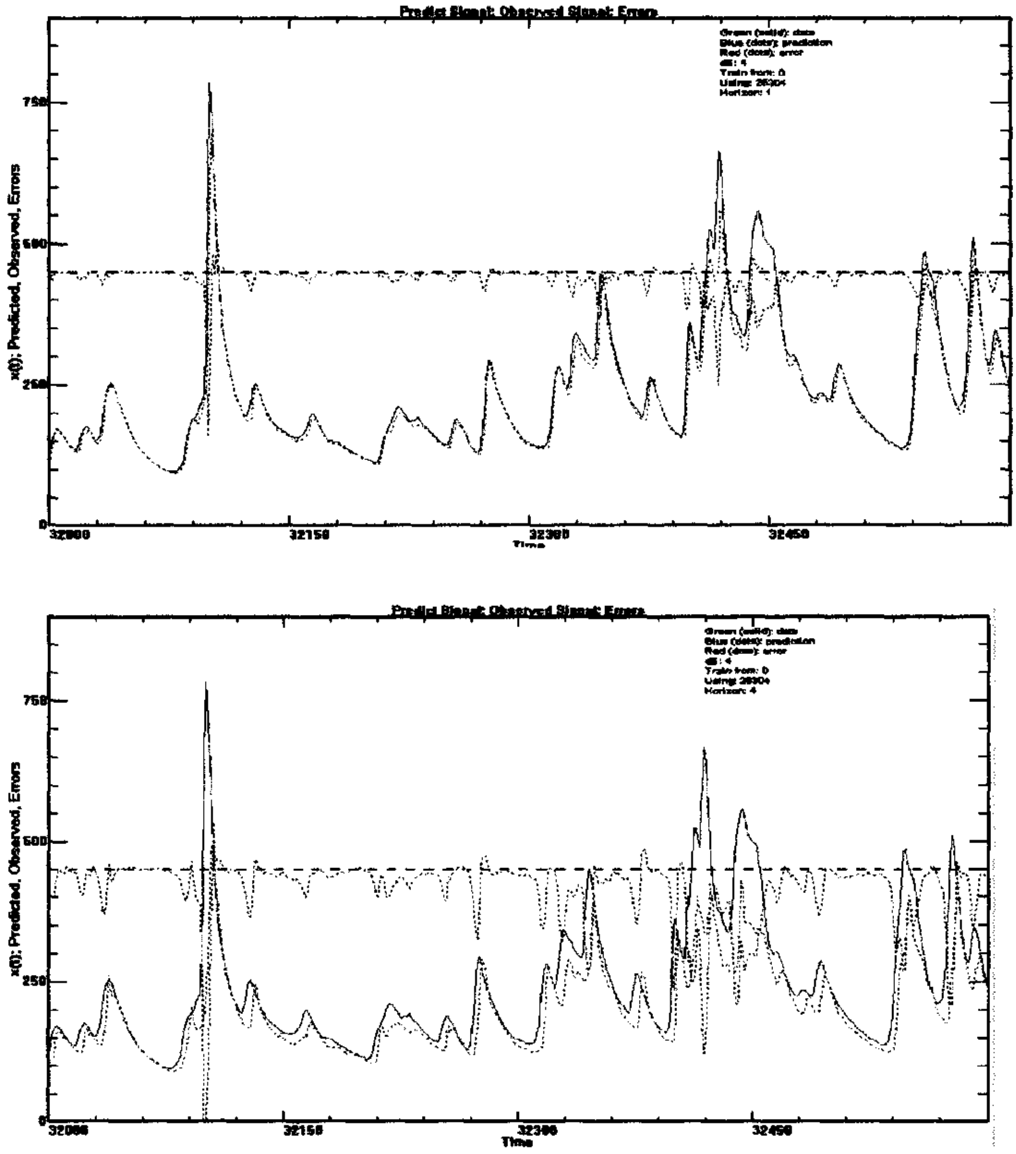

Figure 6-18: Discharge prediction by P(1) model. Medium discharge period. Prediction horizon 1h (upper panel) and 4h (lower panel).

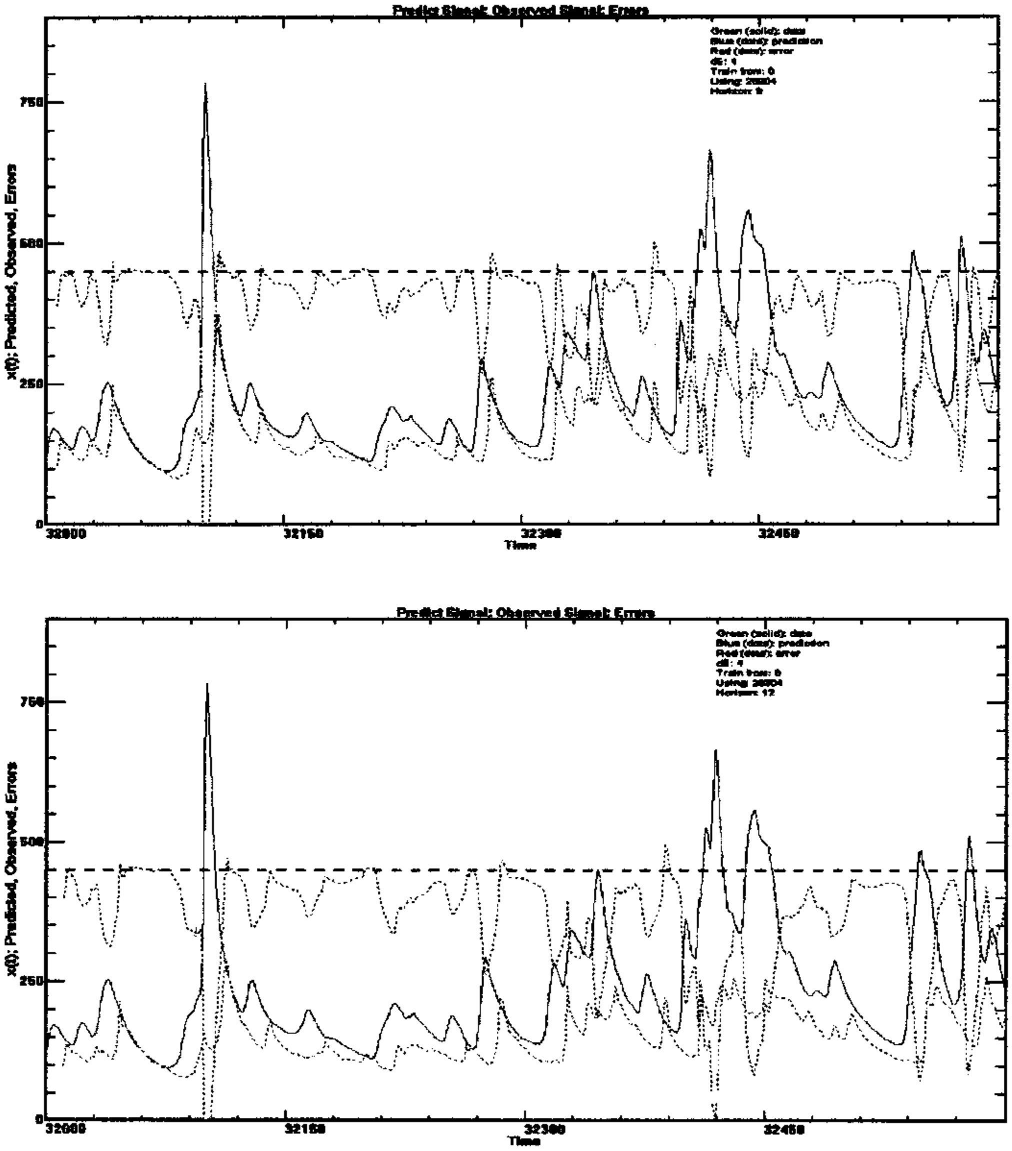

Figure 6-19: Discharge prediction by P(1) model. Medium discharge period. Prediction horizon 8h (upper panel) and 12h (lower panel).

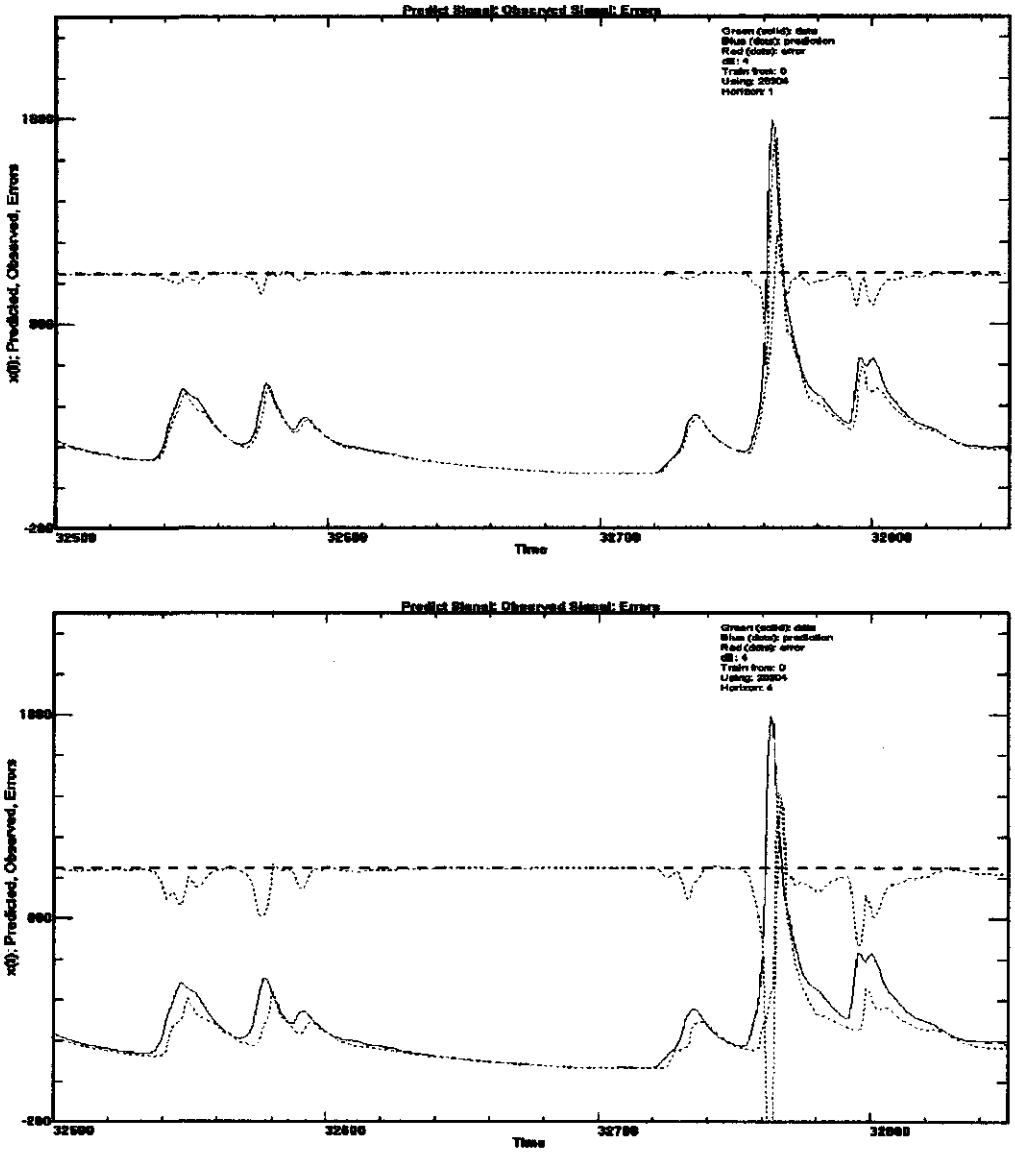

Figure 6-20: Discharge prediction by P(1) model. High discharge period. Prediction horizon 1h (upper panel) and 4h (lower panel).

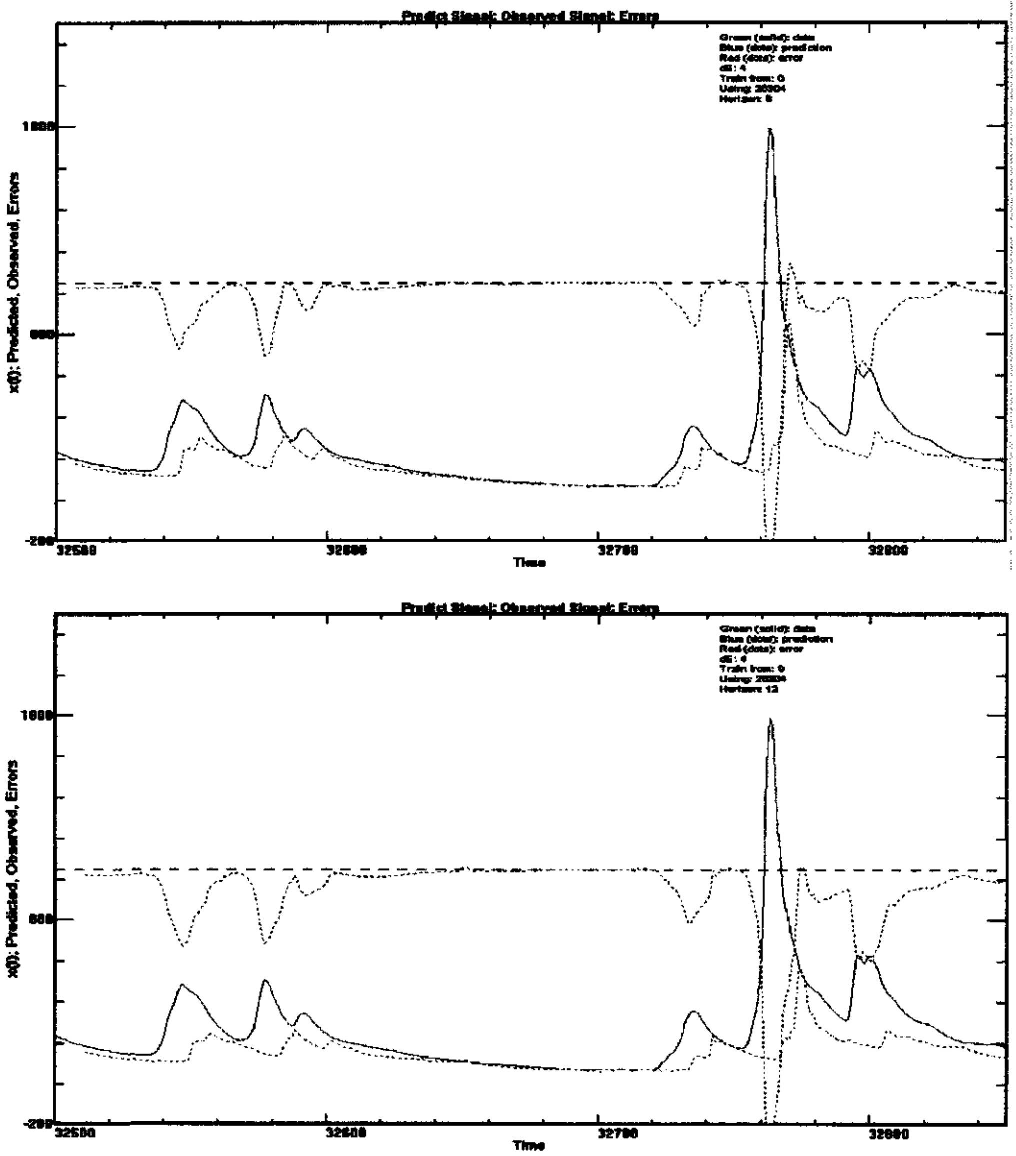

Figure 6-21: Discharge prediction by P(1) model. High discharge period. Prediction horizon 8h (upper panel) and 12h (lower panel).

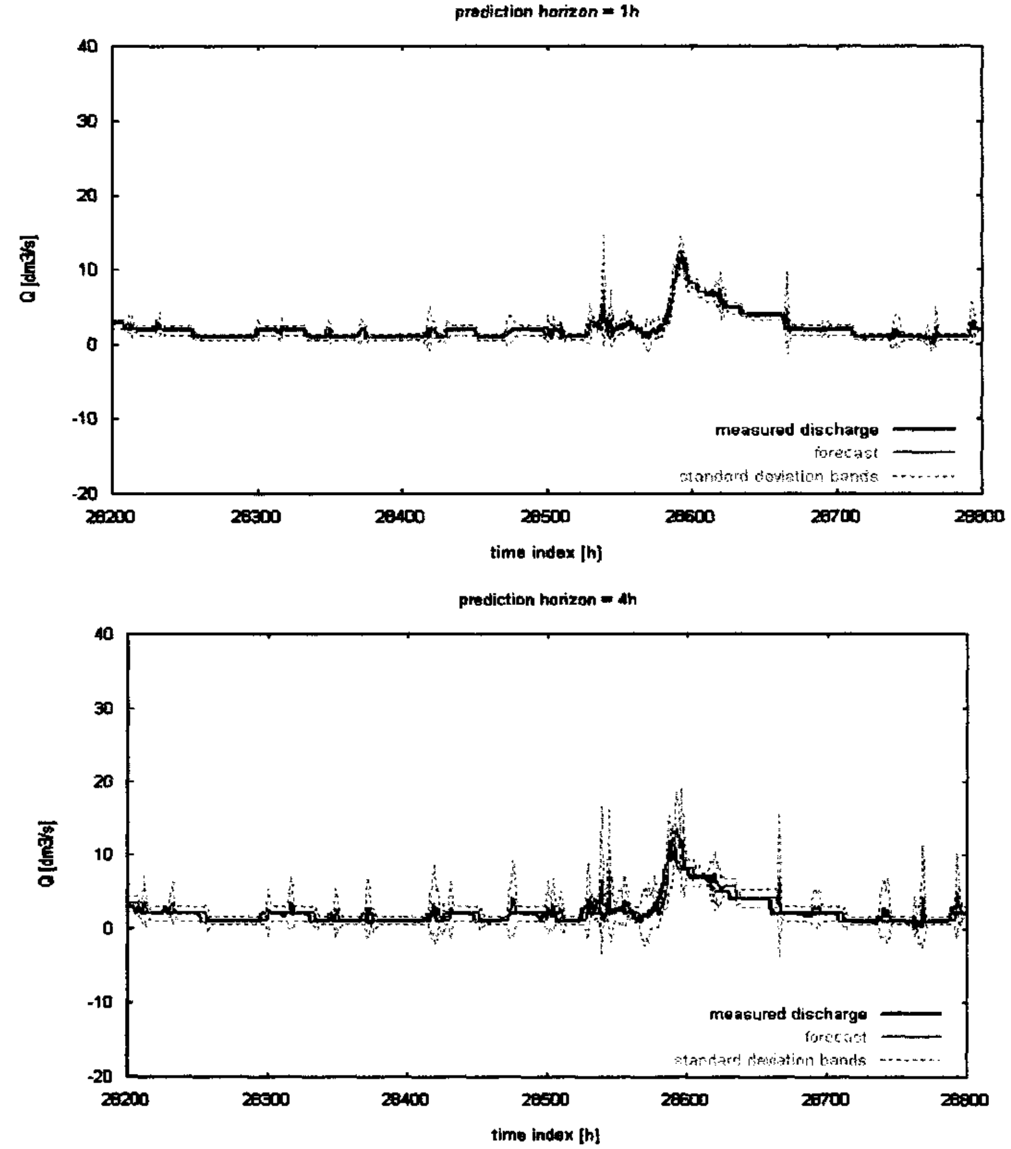

Figure 6-22: LPNN prediction. Low discharge period.

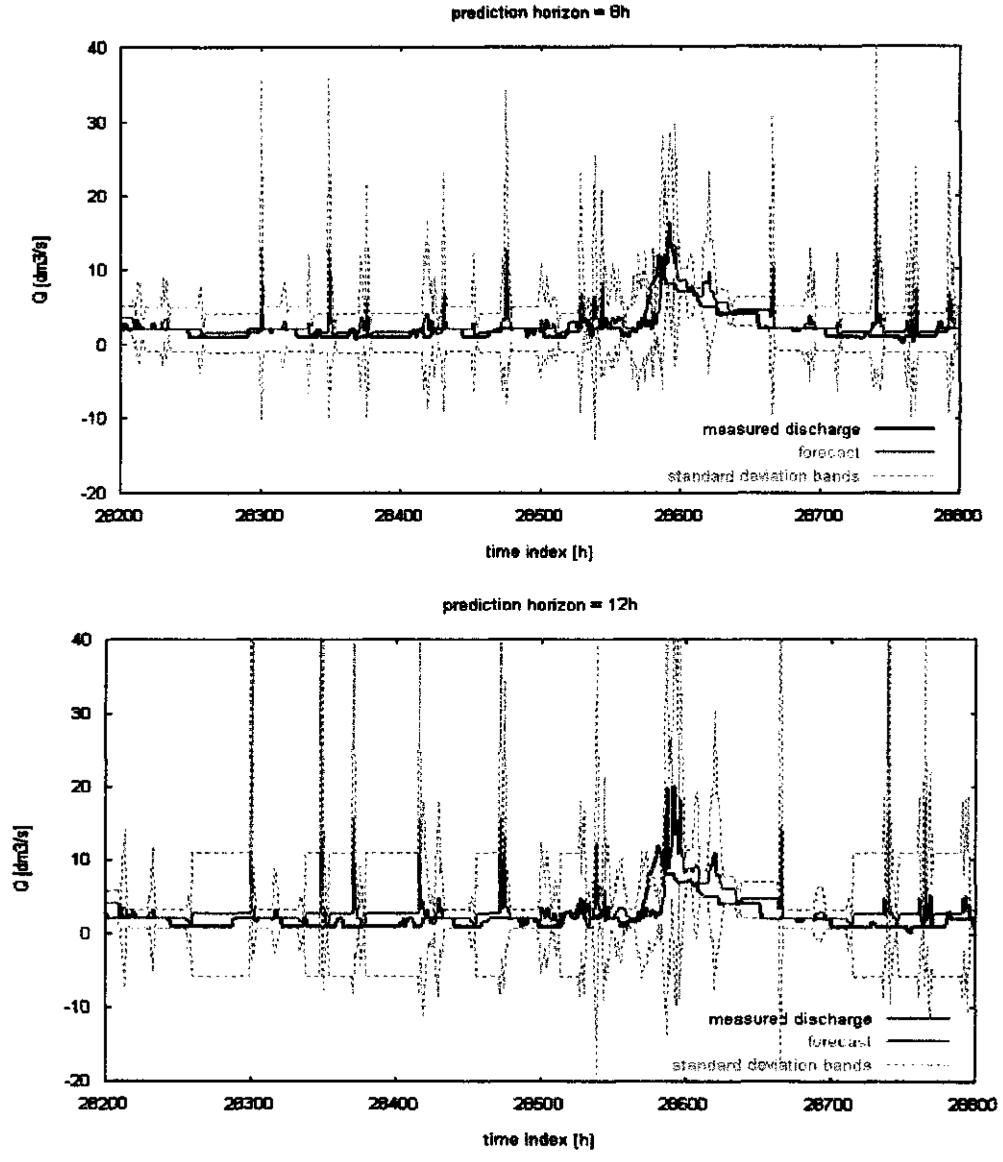

Figure 6-23: LPNN prediction. Low discharge period

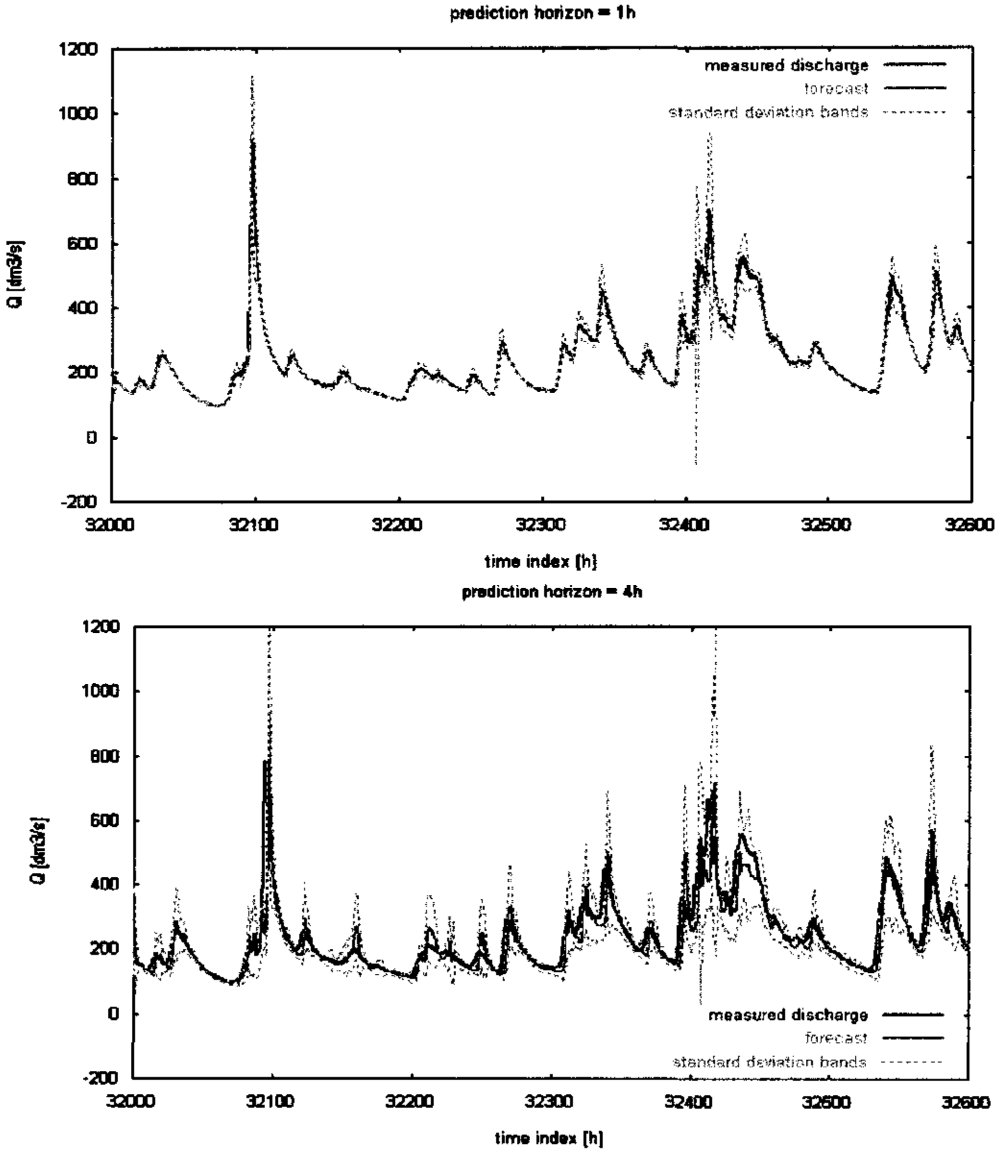

Figure 6-24: LPNN prediction. Medium discharge period.

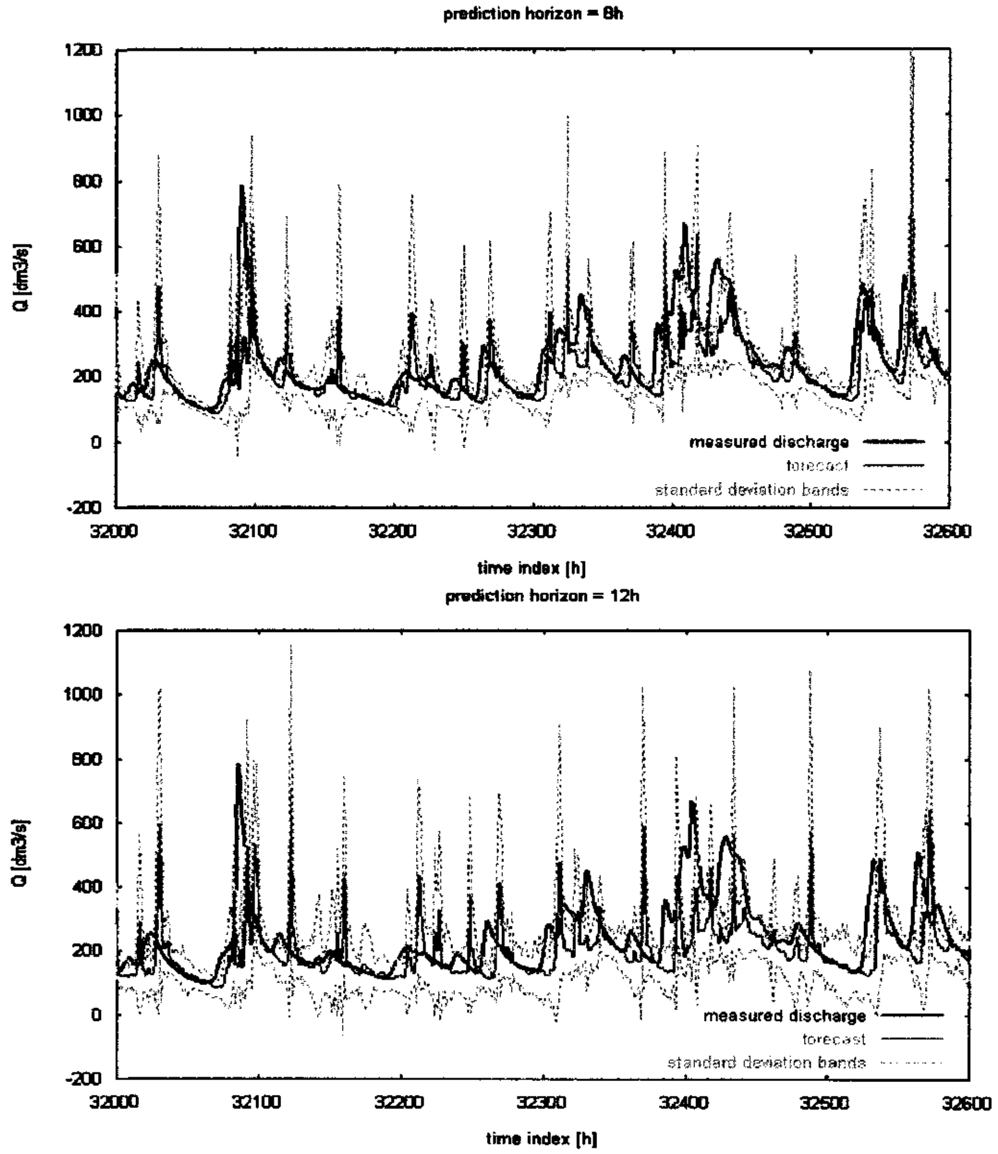

Figure 6-25: LPNN prediction. Medium discharge period.

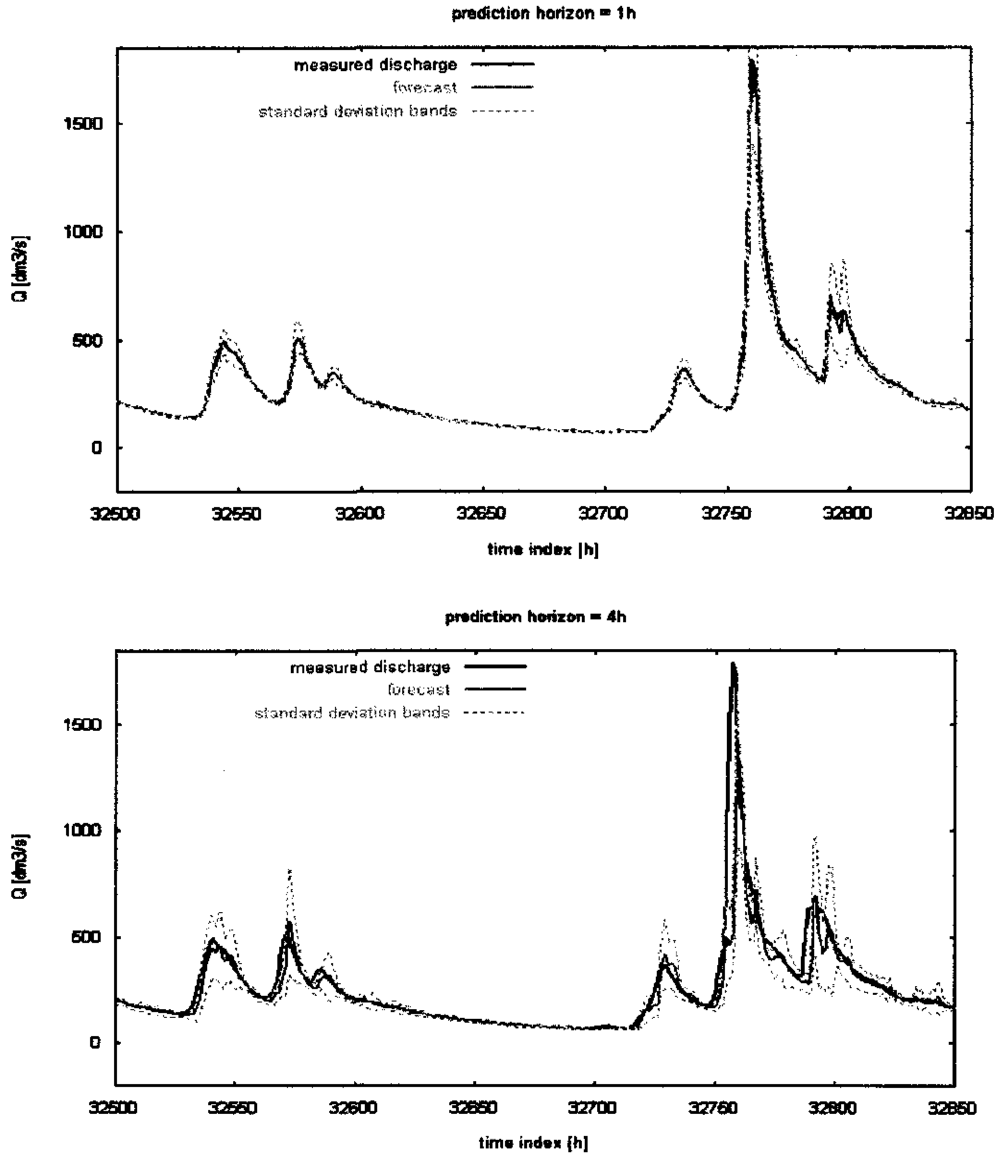

Figure 6-26: LPNN prediction. High discharge period.

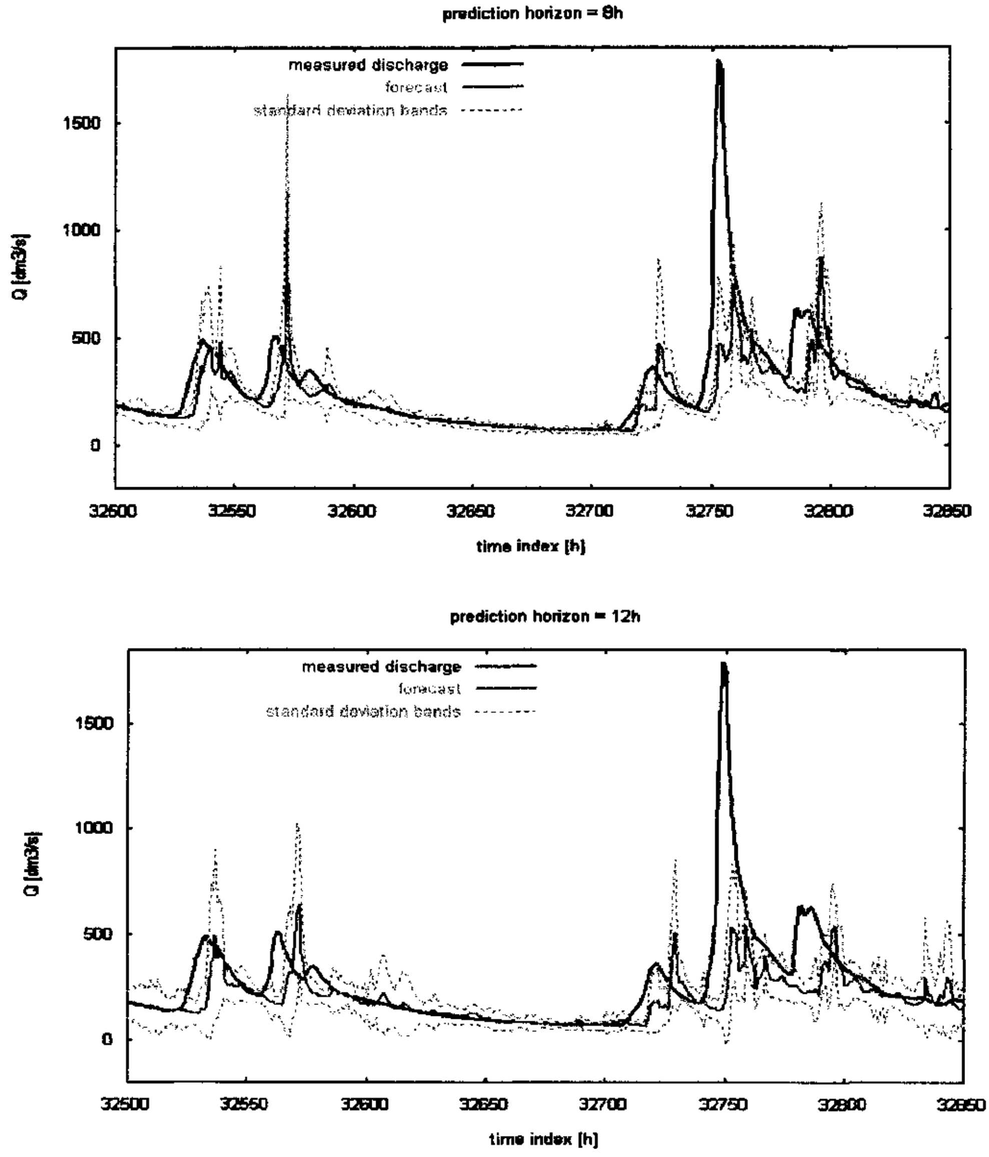

Figure 6-27: LPNN prediction. High discharge period.

next method that has been found very successful was the LPNN. There are three reasons for

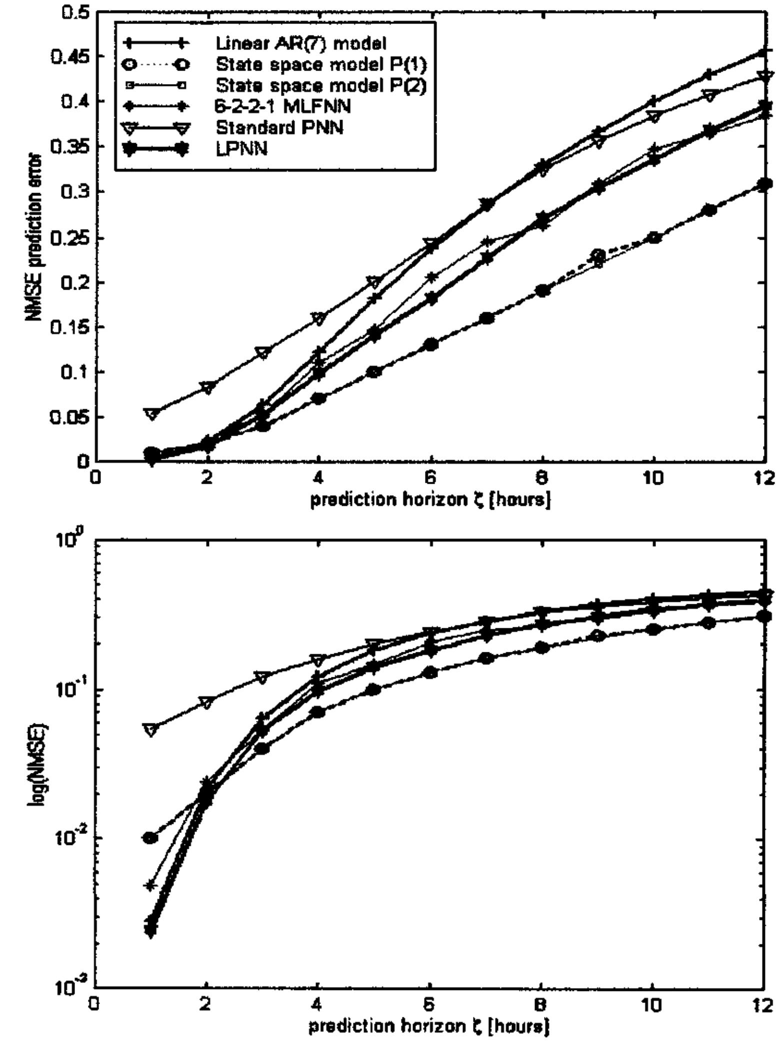

Figure 6-28: Preditability plots for the Hupselse Beek discharge data. Linear-linear (upper panel) and log-linear (lower panel) representations.

this:

- The LPNN was better than AR(7) model for all prediction horizons (see lower panel of Fig. 6-28). This was not the case with any other model. For instance the P(1) and P(2) were doing worse than AR(7) for short prediction horizons: 1 and 2 h ahead.

- Comparing the LPNN and the MLFNN it could be deduced that the LPNN is practically better approximator of the conditional expectation. It can be theoretically shown (Bishop [9]) that the algebraic and probabilistic neural networks should lead to the same prediction results in the least-square sense. Due to unstable training process, however, the MLFNN error curve fluctuated a little which was not observed in the LPNN case.

- Apart from the predictions, the LPNN gives also standard deviation bounds and conditional densities as an extra information for the user. This information is not available from the MLFNN and the state-space models. The trade-off between the probabilistic description of the results and the prediction accuracy is that one has to give away some locality. This would explain why the P(1) and P(2) models surpassed the LPNN for longer prediction times.

The performance of the standard PNN model was rather poor due to the reasons given in Section 4.1.4. The difference in prediction quality between this method and the LPNN confirms once again the strength of the localization concept.

From the analysis of discharge time series from the Hupselse-Beek catchment the decrease of predictive power with increase of prediction horizon has been observed. This decrease scales as shown in lower panel of Fig. 6-28. According to Tsonis et al. [62] such a scaling indicates that the data could be considered as a Fractional Brownian Motion (FBM) series. However, the FBMs are characterized always by normal distributions which was not the case with the Hupselse Beek data. Another possibility could be that this scaling was due to the coexistence of two different dynamic regimes. This would correspond to the results from Section 6.3.1.

Regardless of the nature of the nonlinearity discovered in the studied time series, the interesting question was on which time-scale this nonlinearity occurs. To answer this the AR(7) and the P(1) model were run for ς varying from 1h to 200h. The results were presented as log-log predictability plot (Fig. 6-29). It was observed that the meaningful difference in prediction ability between the two models occurs on the time scale $\cong$ 24h (black spot indicates this).

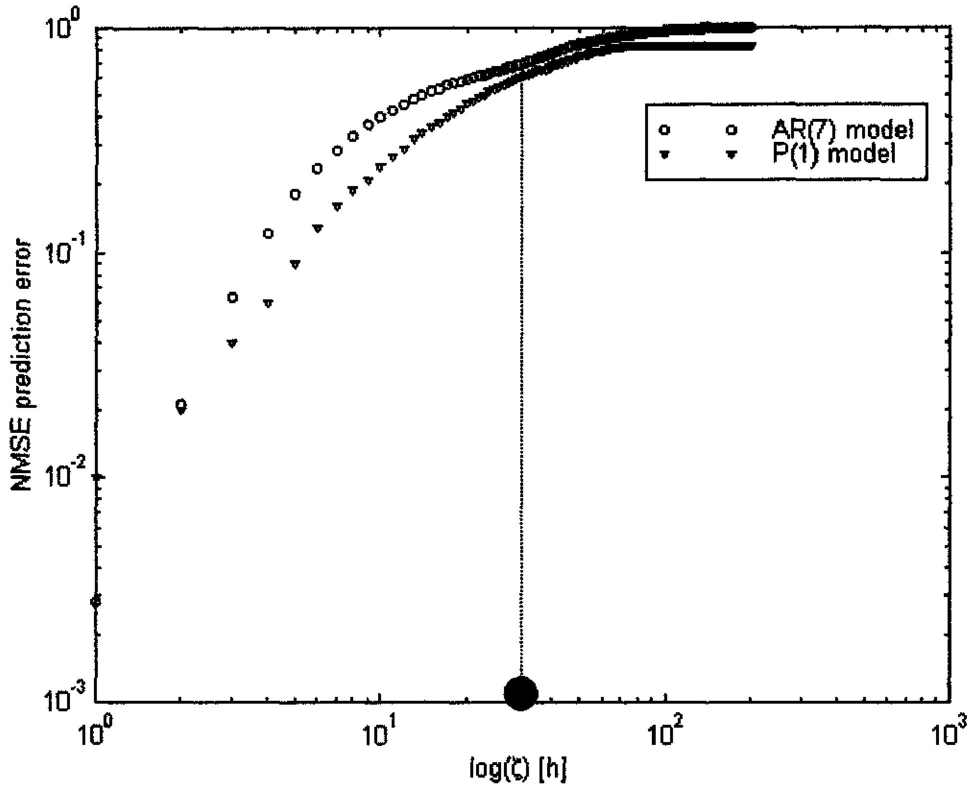

Figure 6-29: Log-log predictability plot for Hupselse Beek discharge. Black spot indicates the time scale of the nonlinearity present in this time series.

Part IV

Synthesis

Chapter 7

Conclusions

As a result of this study, it is clear that both PNNs and state space models proved to be powerful avant-garde discharge prediction techniques. They can be regarded as a superior alternative for the classical linear and non-linear regression models.

In case of the **PNNs** it can be stated that:

1. PNNs form a relatively easy to use tool for modelling input output relationships and for time series prediction.

2. PNN can be easily be made local. This locality gives better results when the data are concentrated on lower dimensional subspaces. This is often the case in higher dimensions. The more inputs a model has, the more likely this is to occur.

3. Besides the prediction, PNN offers other results that can be useful for the modeler : standard deviation, error bounds, full conditional density. All these concepts have the standard probabilistic interpretation. The calculation of these results can be done without much computational effort. This distinguishes Probabilistic Neural Networks from classical feedforward Neural Networks.

4. The conditional densities can be used for simulation.With the help of Monte Carlo integration, this can be employed to calculate the expectation of an arbitrary variable.

5. PNN architecture is completly data-driven. Unlike in the case of feedforward Neural Networks, no difficult a priori decisions have to be taken concerning the structure of the

network.

Summarizing the **state-space methods**, it can be concluded that:

1. Local polynomial models constitute an effective and precise framework for time series prediction
2. They originate from the deterministic view to time series which provides a user with other useful characteristics like global and local embedding dimension. On the other hand, the estimation of these quantities involves some numerical subtleties and requires very skillful programming exertion.
3. For the local polynomial models there is no prediction uncertainty measure built into these algorithms.
4. Calibration of the local polynomials requires large amounts of data to be able to parametrize local neighborhood-to-neighborhood maps correctly. This applies especially when higher order polynomials are used.

Chapter 8

Future research directions

There is still possible work to be done on the LPNNs. The biggest shortcoming of the method is the localization principle (Eq.(4.45) Section 4.1.4). The only reason why this particular scheme was chosen was to avoid computationally expensive and difficult to program nearest neighbor search[1]. No claims are made, however, about optimality and numerical stability of this approach to localization. Moreover, in many practical applications it is required that the outputs in a regression problem should be unchanged, or *invariant*, when the input is subject to linear transformations. In the form presented in this dissertation the LPNNs were not linearly invariant, however, to make those models be so one may think about the following procedure for updating the local covariance matrices:

$$C_j \approx \sum_i e^{-(x_i-x_j)(C_j)^{-1}(x_i-x_j)^T}(x_i - x_j)(x_i - x_j)^T \tag{8.1}$$

Whether this equation has non-trivial solutions is an open research question. Another possible approach would be to treat the input space as a geometric (not parametric) object and define the linear invariance using the framework of differential geometry.

[1] Unlike in the case of the state space models for which the commercial programe was available, the original PNNs C code has been written during this research (see Appendix A).

Appendix A

Software

The calculations were done using the following programs:

- **DPP (Double Precision Parzen)** - a package for simulating the family of Probabilistic Neural Networks developed by Torfs and Wojcik, 1999 in WAU. This will soon be available upon request from the authors as non-commercial software. The core of the package is written in C. As an innovation the LUA 3.1 language is used to allow run-time interaction between LUA programs and their host C kernel. LUA is a free-distribution software available from http:// www.tecgraf.puc-rio.br/ lua
- **CSP for Windows 95/NT** - tools for non-linear time series analysis and prediction. Commercial package available from Applied Nonlinear Sciences, LLC and Randle, Inc. http:// www.zweb.com/ apnonlin/
- **Matlab 5.0** - an integrated technical computing environment that combines numeric computation, advanced graphics and visualization, and a high-level programming language. In particular two additional toolboxes were used : Neural Networks Toolbox and and System Identification Toolbox. For details consult http://www.mathworks.com/

The highly reccomended, public domain graphical package used throughout this Thesis was:

- **GNUPLOT 3.7**. The gnuplot FAQ is available from http://www.uni-karlsruhe.de/~ig25/gnuplot-faq/

The text was typesetted in :

- **Scientific WorkPlace 3.0.** Web site: http://www.tcisoft.com

Appendix B

Neural networks - code examples

MLFNN model for discharge prediction was built with Neural Networks Toolbox for Matlab 5.0. A typical example of computations is presented below.

```
%Discharge modeling by MLFNN
% Rafal Wojcik , WAU
clf reset;
figure(gcf)
colordef(gcf,'none')
setfsize(500,200);
echo off
clc
% ============================
% Creating the MLFNN
% ============================
% INITFF - Initializes a feed-forward network.
% TRAINLM - Trains a feed-forward network with Levenberg-Marquandt scheme.
% SIMUFF - Simulates a feed-forward network.

pause % Strike any key to continue...
clc
% DEFINING A VECTOR ASSOCATION PROBLEM
% ====================================
% P defines input matrix:
load d1hc.dat %calibration set
load d1hv.dat %validation set
Pscale= 2*((d1hc-min(d1hc))/max(d1hc))-1
P = delaysig(Pscale’,7,12)
%T defines the associated target (column vector):
%T = normr(test(:,3)’)
T = Pscale
T=T’
s=size(T)
pause % Strike any key to see these data points...
clc
% PLOTTING THE DATA POINTS
% ========================
% Here the data points are plotted:
plot((1:1:s(:,2)),T(1,:),’b.’);
title(’Training Set’);
xlabel(’time(hours)’);
ylabel(’Q(t)’);
pause
% Strike any key to design the network...
clc
% DESIGN THE NETWORK
% ==================
% A four-layer(6-2-2-1) TANSIG/TANSIG network will be trained.
% The number of hidden TANSIG neurons should reflect the
% complexity of the problem.
S1 = 2;
S2 = 2;
```

```
% INITFF is used to initialize the weights and biases for
% the TANSIG/PURELIN network.
[w1,b1,w2,b2,w3,b3] = initff(P,S1,’tansig’,S2,’tansig’,T,’tansig’);
echo off
echo on
clc
% TRAINING THE NETWORK
% ====================
% TRAINLM uses Levenberg-Marquandt to train the MLFNN networks.
df = 2; % Frequency of progress displays (in epochs).
me = 100; % Maximum number of epochs to train.
eg = 0.00002; % RMS error goal.
mingr = 0.00001; % minimum gradient
tp = [df me eg mingr];
% Training begins...please wait (this takes a while!)...
[w1,b1,w2,b2,w3,b3,ep,tr] = trainlm(w1,b1,’tansig’,w2,b2,’tansig’,w3,b3,’tansig’,P,T,tp);
% ...and finally finishes.
% TRAINLM has returned new weight and bias values, the number
% of epochs trained EP, and a record of training errors TR.
%save c:\results\q\neural\weightsl4t2.dat w1 b1 w2 b2 w3 b3 -ascii
pause
clc
% PLOTTING THE ERROR CURVE
% ========================
% Here the errors are plotted with respect to training epochs:
ploterr(tr,eg);
%pause % Strike any key to use the network ...
clc
% We can now test the network
p=(2*((d1hv-min(d1hc))/max(d1hc))-1)’
```

```
s=size(p)
ptst = delaysig(p,7,12)
a = simuff(ptst,w1,b1,’tansig’,w2,b2,’tansig’,w3,b3,’tansig’)
% The result is fairly close. Training to a lower error
% goal would result in a closer approximation.
plot(1:1:s(:,2),p(1,:),’b.’,1:1:s(:,2),a(1,:),’r.’)
title(’Output prediction’);
xlabel(’time’);
ylabel(’Qreal vs Qpred’);
% error calculation
d1hv=d1hv’
d1hvpred=((a+1)*max(d1hc)/2)+min(d1hc)
resid=d1hv-d1hvpred;
resid=resid(1,13:length(resid))
NMSE=sumsqr(resid)/(length(resid)*var(d1hv))
save c:\results\q\neural\el7r10.dat NMSE -ascii
%d1hvpred=d1hvpred’
%save c:\results\q\neural\pl4t2.dat d1hvpred -ascii
echo off
disp(’end of disch’)
```

The LPNN forecasting method was implemented in the following LUA code :

```
---------------------------------------------
–LPNN discharge prediction by Rafal Wojcik, WAU
---------------------------------------------
execute(”time /T”)
t = VECTOR()
f = readfrom(”d1hcv.dat”)
t:read(f)
calib = t:copy({from=1,to=26304})
verif = t:copy({from=26305,to=35064})
```

```
–make lag vectors
lags = IVECTOR({length=5})
lags[1] = 0
lags[2] = 1
lags[3] = 2
lags[4] = 3
lags[5] = 15
– now read calibration data into a ParzenSet
p = ParzenSet()
p:lagconstruct(lags,calib)
p:undouble()
write(”number of cases in calibration = ”,p.numcases,”\n”)
p.dimY=1

– LPNN prediction –

p:RecalcUVWDE(”localcov”,{alphamode = ”local”,alpha=0.001,beta = 5})
writeto()
write(”ParzenSet set to localcov\n”)
p.bandwith =1
writeto()
write(”calculating conditional moments\n”)
writeto(”l12n5.dat”)
deldisch= VECTOR({length=p.dimX})
i=1
sum=0
while i<=calib.length-lags[lags.length] do
 local k
 k=1
 while k<=p.dimX do
```

```
deldisch[k]=calib[i+lags[k]]
k=k+1
end
m,c = p:condmoments(deldisch)
stdev = sqrt(c[1][1])
obs = calib[i+lags[p.dimX+1]]
write(i,” ”,obs,” ”,m[1],” ”,m[1]+stdev,” ”,m[1]-stdev,”\n”)
sum = sum + (m[1]-obs)*(m[1]-obs)
i=i+1
end
writeto()
write(”MSE =”,” ”, sum/(calib.length-lags[lags.length]),”\n”)
writeto(”l12n5err.dat”)
write(”MSE =”,” ”, sum/(calib.length-lags[lags.length]),”\n”)
writeto()
execute(”time /T”)
quit()
```

Appendix C

UVWDE- decomposition and Parzen densities

C.1 Basic theorem

Let

$$\mathbf{C_{ZZ}} = \begin{bmatrix} \mathbf{C_{XX}} & \mathbf{C_{XY}} \\ \mathbf{C_{YX}} & \mathbf{C_{YY}} \end{bmatrix} \tag{C.1}$$

be a covariance matrix (i.e. a positive definite matrix). Then this matrix can be written as :

$$\mathbf{C_{ZZ}} = \begin{bmatrix} \mathbf{C_{XX}} & \mathbf{C_{XY}} \\ \mathbf{C_{YX}} & \mathbf{C_{YY}} \end{bmatrix} = \begin{bmatrix} \mathbf{UD^2U^\top} & \mathbf{UD^2W^\top} \\ \mathbf{WD^2U^\top} & \mathbf{WD^2W^\top + VE^2V^\top} \end{bmatrix} \tag{C.2}$$

where $\mathbf{D}$ and $\mathbf{E}$ are diagonal matrices and $\mathbf{U}$ and $\mathbf{V}$ are orthonormal, i.e. $\mathbf{U\,U^\top = I}$ and $\mathbf{V\,V^\top = I}$

C.2 Calculation

1. $\mathbf{C_{XX} = UD^2U^\top}$ as before

2. $\mathbf{W = C_{YX}UD^{-2}}$

3. $\mathbf{C_{YY|X} = C_{YY} - WD^2W^\top}$

4. $\mathbf{C_{YY|X}} = \mathbf{VE^2V^\top}$ as before

C.3 Use in simulation

Let ε and η both be (independent) white noise, then the stochast

$$\mathbf{Z} = \begin{bmatrix} \mathbf{X} \\ \mathbf{Y} \end{bmatrix}$$

defined by

$$\begin{aligned} \mathbf{X} &= \mathbf{m_X} + \mathbf{U\,D}\,\varepsilon && \text{(C.3)} \\ \hat{\mathbf{Y}} &= \mathbf{m_Y} + \mathbf{W\,D}\,\varepsilon && \text{(C.4)} \\ \mathbf{Y} &= \hat{\mathbf{Y}} + \mathbf{V\,E}\,\eta && \text{(C.5)} \end{aligned}$$

has mean and covariance given by :

$$\begin{aligned} \mathrm{E}[\mathbf{Z}] &= \mathbf{m_Z} = \begin{bmatrix} \mathbf{m_X} \\ \mathbf{m_Y} \end{bmatrix} && \text{(C.6)} \\ \mathbf{C_{ZZ}} &= \begin{bmatrix} \mathbf{C_{XX}} & \mathbf{C_{XY}} \\ \mathbf{C_{YX}} & \mathbf{C_{YY}} \end{bmatrix} = \begin{bmatrix} \mathbf{UD^2U^\top} & \mathbf{UD^2W^\top} \\ \mathbf{WD^2U^\top} & \mathbf{WD^2W^\top} + \mathbf{VE^2V^\top} \end{bmatrix} && \text{(C.7)} \end{aligned}$$

C.4 Use in conditional simulation

Let $\mathbf{x}$ be given, let η be white noise, then the stochast $\mathbf{Y_x}$ defined by :

$$\begin{aligned} \widehat{\mathbf{Y_x}} &= \mathbf{m_Y} + \mathbf{W\,U^\top}(\mathbf{x} - \mathbf{m_X}) && \text{(C.8)} \\ \mathbf{Y_x} &= \widehat{\mathbf{Y_x}} + \mathbf{V\,E}\,\eta && \text{(C.9)} \end{aligned}$$

has mean and covariance given by :

$$\begin{aligned} \mathrm{E}[\mathbf{Y_x}] &= \mathbf{m_{Y|X=x}} && \text{(C.10)} \\ &= \mathbf{m_Y} + \mathbf{C_{YX}}\mathbf{C_{XX}}^{-1}(\mathbf{x} - \mathbf{m_X}) \end{aligned}$$

$$\begin{aligned} &= \widehat{\mathbf{Y}_\mathbf{x}} \\ \mathrm{COV}[\mathbf{Y}_\mathbf{x}] &= \mathbf{C}_{\mathbf{YY}|\mathbf{X}} \qquad \text{(C.11)} \\ &= \mathbf{C}_{\mathbf{YY}} - \mathbf{C}_{\mathbf{YX}}\mathbf{C}_{\mathbf{XX}}{}^{-1}\mathbf{C}_{\mathbf{XY}} \\ &= \mathbf{V}\mathbf{E}^2\mathbf{V}^\top \end{aligned}$$

C.5 Use in calculation density

If

$$f(\mathbf{z}) = \frac{1}{\sqrt{(2\pi)^N\,|\mathbf{C}_{\mathbf{ZZ}}|}}\ \exp\left[-\frac{\|\mathbf{z}-\mathbf{m}_\mathbf{Z}\|^2{}_{\mathbf{C}_{\mathbf{ZZ}}}}{2}\right] \qquad \text{(C.12)}$$

define $\mathbf{e}_\mathbf{x}$ and $\mathbf{e}_\mathbf{y}$ by :

$$\begin{aligned} \mathbf{e}_\mathbf{x} &= \mathbf{U}^\top(\mathbf{x}-\mathbf{m}_\mathbf{X}) \qquad \text{(C.13)} \\ \widehat{\mathbf{e}}_y &= \mathbf{m}_\mathbf{Y} + \mathbf{W}\,\mathbf{e}_\mathbf{x} \qquad \text{(C.14)} \\ \mathbf{e}_\mathbf{y} &= \mathbf{V}^\top\,\widehat{\mathbf{e}}_y \qquad \text{(C.15)} \end{aligned}$$

then :

$$f(\mathbf{z}) = \prod_{i=1}^{N}\frac{1}{\sqrt{2\pi d_i}}\ \exp\left[-\frac{e^2_{x,i}}{2d_i^2}\right]\ \prod_{i=1}^{M}\frac{1}{\sqrt{2\pi d_i}}\ \exp\left[-\frac{e^2_{y,i}}{2d_i^2}\right] \qquad \text{(C.16)}$$

C.6 Use in calculation of conditional density

Let $\mathbf{x}$ and $\mathbf{y}$ be given. Define :

$$\begin{aligned} \mathbf{y}_\mathbf{x} &= \mathbf{m}_\mathbf{Y} + \mathbf{W}\,\mathbf{U}^\top(\mathbf{x}-\mathbf{m}_\mathbf{X}) \qquad \text{(C.17)} \\ \mathbf{e}_\mathbf{y} &= \mathbf{V}^\top\,(\mathbf{y}-\mathbf{y}_\mathbf{x}) \qquad \text{(C.18)} \end{aligned}$$

then :

$$f_{\mathbf{Y}|\mathbf{X}=\mathbf{x}}(\mathbf{y}) = \prod_{i=1}^{M}\frac{1}{\sqrt{2\pi d_i}}\ \exp\left[-\frac{e^2_{y,i}}{2d_i^2}\right] \qquad \text{(C.19)}$$

Bibliography

[1] Abarbanel H D I (1996) Analysis of Observed Chaotic Data, Springer- Verlag, New York

[2] Abarbanel H D I (1998) Obtaining order in a world of chaos - time domain analysis of nonlinear and chaotic signals, IEEE Signal Processing Magazine, 49-65

[3] Abarbanel H D I, Kennel M B (1993) Local false nearest neighbors and dynamical dimensions from observed chaotic data, Rev. of Mod. Phys., 65(4),1331-1392

[4] Abott M B, Bathurst J C, Cunge J A, O'Connel PE, Rassmusen J (1986) An introduction to the European Hydrological System - Systeme Hydrologique Europeen SHE 2. Structure of the physically based distributed modelling system, Journal of Hydrology 87, 61-77

[5] Adamowski K (1985) Nonparametric kernel estimation of flood frequencies, Water Resour. Res, 21(11), 1585-90

[6] Adamowski K (1989) A Monte Carlo comparison of parametric and nonparametric estimation of flood frequencies, J. Hydrol, 108, 205-308

[7] van den Akker M F A, van de Viel B J H (1996) Hoogwatervoorspellingen vor Rijn bij Lobith met hybride methoden , Report 66, Water Resources Dept., Wegeningen Agricultural University

[8] Babovic V Larsen L C (Ed.) (1998) Hydroinformatics '98, Proceedings of The Third International Conference on Hydroinformatics, Vol. 2, A.A Balkema, Rotterdam

[9] Bishop C M (1995) Neural Networks for Pattern Recognition, Oxford University Press, New York

[10] Blackie J R, Eeles C W O (1985) Lumped catchment models In: Anderson and Burt (Ed.): Hydrological Forecasting, 405-435, Wiley, Chichester

[11] Bonnlander B (1996) Nonparametric selection of input variables for connectionist learning, PhD Thesis, University of Colorado

[12] van den Boogard H F P, Gautam D.K, Mynett A E (1998) Autoregressive neural networks for the modelling of time series, In: Babovic and Larsen (Ed.): Hydroinformatics '98, Proceedings of The Third International Conference on Hydroinformatics, Vol. 2, A.A Balkema, Rotterdam

[13] Box G E P, Jenkins G M (1970) Time Series Analysis, Forecasting and Control, Holden-Day, San Francisco

[14] Casdagli M, Eubank S (Ed.) (1992) Nonlinear modelling and forecasting, Santa Fe Institute Proceedings, Addison Wesley Publ. Co.

[15] Dimopoulos I, Lek S, Lauga J (1996) Rainfall - runoff modelling by neural net- works and Kahnan Filter, Hydrological Sciences Journal 41(2), 179-193

[16] Dooge J C (1958) A general theory of the unit hydrograph, J. Geoph. Res., 64(2)

[17] Edijatno, de Oliviera Nascimento N, Yang X, Makhlouf Z, Michel C (1999) GR3J: a daily watershed model with three free parameters, Hydrological Sciences-Journal-des Sciences Hydrologiques, 44(2), 263-277

[18] Dowla F U, Rogers L L (1995) Solving Problems in Environmental Engineering and Geosciences with Artificial Neural Networks, MIT Press, Cambridge, Massachusetts

[19] Feluch W (1995) Nonparametric estimation of multivariate density and nonparamertic regression, In: Kundzewicz (Ed.): New Uncertainty Concepts in Hydrology and Water Resources, Proceedings of the International Workshop on New Uncertainty Concepts in Hydrology and Water Resources, 145-150, Cambridge University Press, Cambridge

[20] Fraser A M, Swinney H L (1986) Independent coordinates for strange attractors from mutual information, Phys. Rev. A, 33, 1134-1140

[21] Frison T W, Abarbanel D I, Marshall D E, Shulz J R, Scherer W D (1998) Chaos and predictability in ocean water levels, To be published

[22] Gallanger R G (1968) Information Theory and Reliable Communication, John Willey and Sons, New York

[23] Georgakos K P, Sharifi M B, Sturdevant P I (1995) Analysis of high resolution rainfall data, In: Kundzewicz (Ed.): New uncertainty concepts in hydrology and water resources, Proceedings of the International Workshop on New Uncertainty Concepts in Hydrology and Water Resources, 114-119, Cambridge University Press, Cambridge

[24] Gershenfeld N A, Schoner B, Metois E (1998) Cluster weighted modelling for time series prediction and characterization, Nature, vol. 397, 329-332

[25] Gershenfeld N A (1998) The Nature of Mathematical Modelling, Cambridge University Press, Cambridge

[26] Golub G H, Van Loan C F (1989) Matrix computations, The Jons Hopkins, Baltimore and London

[27] Grabs W (Ed.) (1997) Impact of climate change on hydrological regimes and water resources management in the Rhine basin, CHR Report no. I–16

[28] Guo Sheng L (1995) Non-parametric approach for design flood estimation with pre gauging data and information, In: Kundzewicz (Ed.): New uncertainty concepts in hydrology and water resources, Proceedings of the International Workshop on New Uncertainty Concepts in Hydrology and Water Resources, 145-150, Cambridge University Press, Cambridge

[29] Hanish W S, Pires E C, Carvalho A C P L F (1998) A neural network model to predict operational parameters of a wastewater treatment plant, In: Babovic and Larsen (Ed.): Hydroinformatics '98, Proceedings of The Third International Conference on Hydroinformatics, Vol. 2, A.A Balkema, Rotterdam

[30] Hall M J, Minns A W (1998) Regional flood frequency analysis using artificial neural networks, In: Babovic and Larsen (Ed.): Hydroinformatics '98, Proceedings of The Third International Conference on Hydroinformatics, Vol. 2, A.A Balkema, Rotterdam

[31] Hsu-KuoLin, Gupta HV, Sorooshian S (1995) Artificial neural network modelling of the rainfall-runoff process, Water Resour. Res. 31(10), 2517-2530

[32] Kanz H, Schreiber T (1997) Non-linear Time Series Analysis, Cambridge University Press

[33] Kennel MB, Brown R, Abarbanel HDI (1992) Determining embedding dimensions for phase-space reconstruction, Phys.Rev. A,45, 3403-3411

[34] Kurkova V (1992) Kolmogorov's theorem and multilayer networks, Neural Networks, 5(3), 501-506

[35] Lall U, Moon Y, Bosworth K (1993) Kernel flood frequency estimators: bandwidth selection and kernel choice, Water Resour. Res., 29(4), 1003-1015

[36] Lorenz E N (1963) Deterministic nonperiodic flow, J.Atmos.Sci., 20, 130 -141

[37] Lorrai M, Sechi GM (1995) Neural nets for modelling rainfal-runoff transformations, Water Resources Management, 9, 299-313

[38] Lula P, Pociask-Karteczka J (1996) Zastosowanie sieci neuronowych do okreslania wielkosci parowania, Prz. Geof.,4, 311-321

[39] Masters T (1995a) Advanced Algorithms for Neural Networks - a C++ source book, John Willey & Sons, New York

[40] Masters T (1995b) Neural, Novell & Hybrid Algorithms for Time Series Prediction, John Willey & Sons, New York

[41] Minns AW, Hall MJ (1996) Artfficial neural networks as rainfall - runoff models, Hydrological Sciences Journal 41(3), 399-417

[42] Nash J E (1960) A unit hydrograph study with particular reference to British Catchments, Proc. Inst. Civ. Eng., 17(5)

[43] Nemec J (1986) Design and operation of forecasting operational real time hydrological systems (FORTH), In: Kraijenhoff D and Moll E (Ed.): River flow modelling and forecasting, 299-322, D.Reidel Publishing Company, Dordrecht

[44] Palus M (1994) Identifying and quantifying chaos by using information-theoretic functionals, In: Weigend A and Gershenfeld N (Ed.): Time Sseries Prediction : Forecasting the Future and Understanding the Past, 387-413, Addison Weseley, Reading, Massachusets

[45] Parzen E (1962) On estimation of a probability density function and mode, Annals of Mathematical Statistics, 33, 1065-1076

[46] Powell M J D (1992) The theory of radial basis function approximation in 1990, In: Light and Will (Ed.): Advances in numerical analysis, vol.II,105-210, Oxford University Press, Oxford

[47] Press W H, Flannery B P, Teukolsky S A, Vetterling W T (1989) Numerical Recipes in C - The Art of Scientific Computing, Cambridge University Press, Cambridge

[48] Priestley (1981) Spectral Analysis and time Series, pp. 867-868, Academic Press, London

[49] Rajagopalan B, Lall U, Torboton D G (1997) Evaluation of kernel density methods for daily precipitation resampling, Stochastic Hydrol. Hydraul. 11(6), 523-54

[50] Rao T S, Gabr M M (1984) An Introduction to Bispectral Analysis and Bilinear time Series Models, In: Brilinger, Fienberg, Gani, Hartigan, Krickberg (Ed.): Lecture Notes in Statistisc, vol.24, Springer-Verlag, New York

[51] Shanon C E, Weaver W (1949) The Mathematical Theory of Communication , University of Illinois Press, Urbana

[52] Sharma A, Lall U, Tarboton D G (1998) Nonparametric streamflow simulation model, Stochastic Hydrol. Hydraul. 12(1), 33-52

[53] Silverman BW (1986) Density Estimation for Statistics and Data analysis, Chapman nad Hall, New York

[54] Specht D (1991) A generalized regression neural network, IEEE Transactions on Neural Networks, 2(6), 568-576

[55] Stam C J, Pijn J P M, Pritchard W S (1998) Reliable detection of nonlinearity in experimental time series with strong periodic components, Physica D, 112, 361-380

[56] Schreiber T , Schmitz A (1996) Improved surrogate data for nonlinearity tests, Phys. Rev. Lett. 77(4)

[57] Schreiber T (1998) Constrained randomization of time series data, Phys. Rev. Lett. 80(10)

[58] Takens F (1981) Detecting strange attractors in turbulence, In: Rand and Young (Ed.): Dynamical systems and turbulence, Warwick, Berlin, Springer

[59] Theiler J, Eubank S, Longtin A, Galdrikian B, Farmer J D (1992) Testing for nonlinearity in time series: The method of surrogate data, Physica D 58(77)

[60] Tong H (1995) Non-linear Time Series - a Dynamical Systems Approach, Oxford University Press, New York

[61] Torfs P, Bier G (1999) A new technique in parametrizing and calibrating models (to be published)

[62] Tsonis A A, Elsner J B (1988) Nonlinear prediction as a way of distinguishing chaos from random fractal sequences, Nature, vol. 358, 217-220

[63] Wand M P, Jones M C(1995) Kernel Smoothing, Monographs on statistics and applied probability (60), Chapman & Hall, London

[64] Watts G (1997) Hydrological modelling in practice, In: Wilby (Ed.): Contemporary Hydrology - Towards Holistic Environmental Science, John Willey & Sons, Chichester, New York

[65] Wojcik R (1995) Input selection for neural networks by genetic algorithms, M.Sc Thesis, Warsaw Agricultural University

[66] Zaldivar J M, Strozzi F, Gutierrez E, Shepherd I M, Tomasin A (1998) Early detection of high waters at Venice Lagoon using chaos theory, In: Babovic and Larsen (Ed.): Hydroinformatics '98, Proceedings of The Third International Conference on Hydroinformatics, Vol. 2, A.A Balkema, Rotterdam