TKK Dissertations in Information and Computer Science
Espoo 2008 TKK-ICS-D11

INPUT VARIABLE SELECTION METHODS FOR CONSTRUCTION OF INTERPRETABLE REGRESSION MODELS

Jarkko Tikka

Dissertation for the degree of Doctor of Science in Technology to be presented with due permission of the Faculty of Information and Natural Sciences for public examination and debate in Auditorium T1 at Helsinki University of Technology (Espoo, Finland) on the 12th of December, 2008, at 12 o'clock noon.

Helsinki University of Technology
Faculty of Information and Natural Sciences
Department of Information and Computer Science

Teknillinen korkeakoulu
Informaatio- ja luonnontieteiden tiedekunta
Tietojenkäsittelytieteen laitos

Distribution:
Helsinki University of Technology
Faculty of Information and Natural Sciences
Department of Information and Computer Science
P.O. Box 5400
FI-02015 TKK
FINLAND
Tel. +358 9 451 3272
Fax +358 9 451 3277
E-mail: series@ics.tkk.fi

ISBN 978-951-22-9663-7 (Print)
ISBN 978-951-22-9664-4 (Online)
ISSN 1797-5050 (Print)
ISSN 1797-5069 (Online)
URL: http://lib.tkk.fi/Diss/2008/isbn9789512296644

Multiprint Oy
Espoo 2008

Tikka, J. (2008): **Input variable selection methods for construction of interpretable regression models**. Doctoral thesis, Helsinki University of Technology, Dissertations in Information and Computer Science, TKK-ICS-D11, Espoo, Finland.

Keywords: data analysis, machine learning, function approximation, multiresponse linear regression, nonlinear regression, artificial neural networks, input variable selection, model selection, constrained optimization

ABSTRACT

Large data sets are collected and analyzed in a variety of research problems. Modern computers allow to measure ever increasing numbers of samples and variables. Automated methods are required for the analysis, since traditional manual approaches are impractical due to the growing amount of data. In the present thesis, numerous computational methods that are based on observed data with subject to modelling assumptions are presented for producing useful knowledge from the data generating system.

Input variable selection methods in both linear and nonlinear function approximation problems are proposed. Variable selection has gained more and more attention in many applications, because it assists in interpretation of the underlying phenomenon. The selected variables highlight the most relevant characteristics of the problem. In addition, the rejection of irrelevant inputs may reduce the training time and improve the prediction accuracy of the model.

Linear models play an important role in data analysis, since they are computationally efficient and they form the basis for many more complicated models. In this work, the estimation of several response variables simultaneously using the linear combinations of the same subset of inputs is especially considered. Input selection methods that are originally designed for a single response variable are extended to the case of multiple responses. The assumption of linearity is not, however, adequate in all problems. Hence, artificial neural networks are applied in the modeling of unknown nonlinear dependencies between the inputs and the response.

The first set of methods includes efficient stepwise selection strategies that assess usefulness of the inputs in the model. Alternatively, the problem of input selection is formulated as an optimization problem. An objective function is minimized with respect to sparsity constraints that encourage selection of the inputs. The trade-off between the prediction accuracy and the number of input variables is adjusted by continuous-valued sparsity parameters.

Results from extensive experiments on both simulated functions and real benchmark data sets are reported. In comparisons with existing variable selection strategies, the proposed methods typically improve the results either by reducing the prediction error or decreasing the number of selected inputs or with respect to both of the previous criteria. The constructed sparse models are also found to produce more accurate predictions than the models including all the input variables.

Tikka, J. (2008): **Syötemuuttujanvalintamenetelmiä tulkittavien regressiomallien konstruoimiseen**. Tohtorin väitöskirja, Teknillinen korkeakoulu, Dissertations in Information and Computer Science, TKK-ICS-D11, Espoo, Finland.

Keywords: data-analyysi, koneoppiminen, funktion approksimointi, lineaarinen monivasteregressio, epälineaarinen regressio, keinotekoiset hermoverkot, syötemuuttujanvalinta, mallinvalinta, rajoitettu optimointi

TIIVISTELMÄ

Suuria tietoaineistoja kerätään ja analysoidaan monenlaisissa tutkimusongelmissa. Modernit tietokoneet mahdollistavat mitattavien havaintojen ja muuttujien lukumäärän jatkuvan lisääntymisen. Analysoinnissa tarvitaan automaattisia menetelmiä, koska perinteiset manuaaliset menettelytavat ovat käyttökelvottomia kasvavien tietoaineistojen takia. Tässä väitöskirjassa esitetään lukuisia mallinnusoletuksien puitteissa havaittuihin tietoaineistoihin perustuvia laskennallisia menetelmiä tuottamaan käyttökelpoista tietoa datan kuvaamasta systeemistä.

Syötemuuttujanvalintamenetelmiä esitetään sekä lineaaristen että epälineaaristen funktioiden approksimointitehtävissä. Muuttujanvalinta on saavuttanut yhä enemmän huomiota monissa sovelluksissa, koska se auttaa alla olevan ilmiön tulkitsemisessa. Valitut muuttujat korostavat ongelman kannalta merkityksellisimpiä ominaisuuksia. Merkityksettömien muuttujien hylkääminen voi lisäksi lyhentää mallin opetusaikaa ja parantaa sen ennustustarkkuutta.

Lineaarisilla malleilla on tärkeä rooli data-analyysissä, koska ne ovat laskennallisesti tehokkaita ja ne luovat pohjan monille monimutkaisemmille malleille. Väitöskirjassa tarkastellaan erityisesti samanaikaista usean vastemuuttujan estimointia käyttäen samojen syötteiden lineaarikombinaatioita. Muuttujanvalintamenetelmät, jotka ovat alun perin suunniteltu yhdelle vastemuuttujalle, laajennetaan monivastetapaukseen. Oletus lineaarisuudesta ei kuitenkaan ole perusteltu kaikissa ongelmissa. Siksi syötteiden ja vasteen välisten tuntemattomien epälineaaristen riippuvuuksien mallintamisessa sovelletaan keinotekoisia hermoverkkoja.

Ensimmäinen menetelmien joukko sisältää tehokkaat askelettaiset valintastrategiat, jotka määrittävät muuttujien hyödyllisyyden mallissa. Vaihtoehtoisesti muuttujanvalintatehtävä formuloidaan optimointitehtävänä. Kohdefunktiota minimoidaan syötteiden valintaa kannustavan harvuusrajoitteen suhteen. Ennustustarkkuuden ja valittujen muuttujien lukumäärän välistä kompromissia säädellään jatkuva-arvoisella harvuusparametrilla.

Työssä raportoidaan tulokset laajoista kokeista, joissa on käytetty sekä simuloituja funktioita että todellisia testiaineistoja. Olemassa oleviin muuttujanvalintamenetelmiin verrattaessa esitetyt menetelmät parantavat tyypillisesti tuloksia pienentämällä ennustusvirhettä, valitsemalla vähemmän muuttujia tai kummallakin mainitulla tavalla. Lisäksi konstruoitujen harvojen mallien todetaan tuottavan tarkempia ennustuksia kuin mallien jotka sisältävät kaikki syötemuuttujat.

Preface

This work has been carried out at the Laboratory of Computer and Information Science (the Department of Information and Computer Science since January 1st, 2008) of Helsinki University of Technology (TKK). The work in the department and in the laboratory has been funded by the Adaptive Informatics Research Centre (AIRC) and earlier by the Neural Networks Research Centre. This work has also been funded by the Graduate School of Computer Science and Engineering of TKK and the Academy of Finland in the research project: Analysis of dependencies in environmental time series data. Moreover, the personal grants awarded by Emil Aaltonen foundation and Finnish Foundation for Technology Promotion (TES) are acknowledged.

I am grateful to my supervisor Prof. Olli Simula and my instructor Dr. Jaakko Hollmén for their excellent guidance and endless encouragement during the years. They have trusted my judgment and allowed me to choose the research topics according to my personal interests. I thank also the director of AIRC Prof. Erkki Oja for providing the first-class facilities to do research.

I am indebted to my co-authors of the publications of this thesis Dr. Jaakko Hollmén and Dr. Timo Similä. Their contribution is evident and highly appreciated. I am also thankful to all my previous and current colleagues that have created stimulating and sociable atmosphere in the laboratory. Especially, sharing the office room with Dr. Timo Similä, Dr. Mika Sulkava, and Dr. Pasi Lehtimäki was an inspiring and pleasurable experience that I will never forget.

I thank the pre-examiners Prof. Michel Verleysen and Dr. Patrik O. Hoyer for reviewing this thesis and providing valuable feedback, and Prof. Colin Fyfe for agreeing to be the opponent in the defense.

I express my gratitude to my parents and my sister for always supporting me in all my endeavors. I am also grateful to my splendid friends for all the fun moments. Finally, I want to thank my beloved Marja-Leena.

Otaniemi, Espoo, November 18, 2008

Jarkko Tikka

Contents

Abbreviations and notations

AW-RBF	adaptively weighted radial basis function
CV	cross-validation
LARS	least angle regression
LASSO	least absolute shrinkage and selection operator
MLP	multilayer perceptron
MRSR	multiresponse sparse regression
MSE	mean of squared errors
OLS	ordinary least squares
RBF	radial basis function
RR	ridge regression
SISAL	sequential input selection algorithm
SSE	sum of squared errors
SVS	simultaneous variable selection
$\boldsymbol{X}$	$N \times d$ input data matrix
$\boldsymbol{Y}$	$N \times q$ output data matrix
$\boldsymbol{E}$	$N \times q$ matrix of random errors

In the matrices $\boldsymbol{X}$, $\boldsymbol{Y}$, and $\boldsymbol{E}$, the columns denote the variables and the rows corresponds to the observations.

$\boldsymbol{W}$	$d \times q$ matrix of regression coefficients, columns $\boldsymbol{w}_i$ and rows $\underline{\boldsymbol{w}}_i$
$\mathcal{A}$	set of active inputs, i.e. nonzero rows of the matrix $\boldsymbol{W}$
$\boldsymbol{I}$	identity matrix
$\boldsymbol{\beta}$	regression coefficients of a single response linear regression model
$\boldsymbol{x}$	vector of observations of d variables $\boldsymbol{x} = [x_1, \ldots, x_d]$
$\boldsymbol{w}$	vector of adjustable parameters
$f(\boldsymbol{x}, \boldsymbol{w})$	real-valued function depending on variables $\boldsymbol{x}$ and parameters $\boldsymbol{w}$
ε	random error
y	an observation of a one-dimensional output variable
$\hat{y}$	an estimate of the observation y
$p(\boldsymbol{x})$	probability density function
$p(\lVert \cdot \rVert_2)$	penalty function
t, r	nonnegative shrinkage / tuning parameters
λ	nonnegative regularization parameter
$\phi(\cdot)$, $\varphi(\cdot)$	activation functions of neural networks
$\psi(\boldsymbol{x})$	a mapping from the input space to a high-dimensional feature space
$K(\boldsymbol{x}, \boldsymbol{y})$	kernel / basis function
$K_{\boldsymbol{w}}(\boldsymbol{x}, \boldsymbol{y})$	adaptively weighted basis function
L_α, $\lVert \cdot \rVert_\alpha$	α-norm of a vector, $\alpha \geq 1$
$\lVert \cdot \rVert_F$	Frobenius norm of a matrix

Chapter 1

Introduction

1.1 Scope of the thesis

Our world is filled with data. Nowadays, it is common to collect information from many different sources and phenomena, for instance, astronomical objects, environment, economics, industrial processes, and functioning of the human genome. Sizes of databases have dramatically increased in a short period of time, which has boosted interest toward effective data analysis methods. Extraction of unsuspected and useful novel information about the system that has generated the data is known as data mining (Hand et al., 2001). Data mining combines statistics and computer science utilizing advances in both disciplines. Learning from data relies on statistical models and modern computers allow analyses of huge databases (Hastie et al., 2001).

Information can be presented in many ways. A data set may include, for example, images, text, audio, numerical measurements, answers to questionnaires, or any combination of them. For the analysis purposes, it is typically required that the data are represented in a numerical format. Qualitative data, such as a degree of satisfaction, cannot be precisely defined but it can be approximated using numerical values. In a quantitative data set, variables describe specific quantities and can therefore be exactly measured. Quantitative variables are expressed on continuous or discrete scales including nominal and categorical data as well.

In the present thesis, only quantitative data are considered. It is also assumed, that data are presented by two matrices of size $N \times d$ and $N \times q$, where N is the number of samples or observations, d is the number of input variables, and q is the number of output or response variables. The matrices are referred as input and output data, respectively. Advances in measurement techniques and storage capacities have led to the proliferation of the number of input variables and observations in various applications. For instance, several thousands of input variables are encountered in bioinformatics (Hastie and Tibshirani, 2004; Kiiveri, 2008) and hundreds of thousands of observations are processed in knowledge discovery from text documents (Mooney and Bunescu, 2007).

The present thesis concentrates on regression problems, i.e. on the learning of an input-output mapping from observed data (Cherkassky and Mulier, 1998). The obtained models should fulfill two goals. First, they should be able to predict accurately output values for novel observations of the input variables. The second goal is to find the most important or informative subset of input variables. In other words, the aim is to reduce the input space dimension such that the generalization capability of the constructed model is maximal, which corresponds to the problem of finding the most relevant input variables (George, 2000). Parsimonious and compact representations of the data allow easier interpretation of the underlying phenomenon.

The problem of input selection is difficult because of the combinatorial complexity (Kohavi and John, 1997). Consider the problem of finding a subset of $k = 10$ inputs from the $d = 50$ available inputs such that the prediction error is minimized. The exhaustive search is impossible even with the most simplest models, since $d!/(k!(d-k)!) > 10^{10}$ models should be evaluated. If k was not known in advance, the number of possible subsets would be $2^d > 10^{15}$. The computational methods that are proposed in this thesis approximate the exhaustive search. Both less complex search strategies and transformations of the input selection process into a single optimization problem are introduced.

The linear regression models are attractive in many cases because of computational efficiency and interpretability (Draper and Smith, 1981). The regression coefficients can be expressed analytically in the closed form, if the classical sum of squared errors between the observations and the model outputs is minimized. Also, statistical significance tests for several properties of the model are readily available. However, the interpretation of the models may be misleading in the presence of correlated inputs, since the contribution of an input can be compensated by other ones in that case. Misinterpretations caused by the correlated inputs can be reduced by input selection procedures. In the case of multiple responses, modeling could be performed individually for each response or simultaneously with a single model. Although all the separate models would be sparse, it does not necessarily cause a reduction in the input space dimension. In this thesis, several input selection algorithms for the simultaneous estimation of several responses are proposed.

Obviously, all problems cannot be solved by linear models. Although it would be known that dependency between the inputs and the output is nonlinear, the obstacle is that the actual underlying functional form is typically completely unknown. Artificial neural networks (ANNs) are appropriate for modeling nonlinear dependencies (Bishop, 1995; Haykin, 1999), since they are capable to approximate a wide class of functions very well without restrictive assumptions about the data generating process (Hornik et al., 1989). Historically, ANNs can be seen as biologically inspired systems, but in this work they are considered as computational methodologies to solve problems with little a priori information. The major deficiency of neural networks is their black box characteristics, that is, importance of an individual input variable is not clearly expressed and even relative importance of the inputs in the prediction task are hard to assess.

In the present thesis, numerous input selection methods are proposed to improve interpretability of ANN models. Non-informative inputs are discarded from the final network and relative importance of the selected inputs are highlighted. Generalization performance of the network depends on its complexity that can be controlled by varying the number of neurons or input variables. In addition to stepwise selection procedures, optimization strategies to adjust complexity of an initially oversized network are presented. In optimization approaches, the final numbers of neurons and inputs are determined by tuning continuous-valued hyperparameters.

1.2 Contributions presented in the thesis

The present thesis contains the following scientific contributions:

- Variable selection methods are proposed to produce interpretable and accurate linear models for the simultaneous prediction of multiple responses. The methods set some of the regression coefficients exactly to zero and shrink the rest toward zero.
- For the selection and shrinkage in the multiresponse linear regression, variants of the forward stepwise algorithm and a constrained optimization problem are introduced. The proposed methods extend the LARS algorithm and the LASSO formulation to the case of multiple responses. In addition, an efficient algorithm to follow the entire solution path of the constrained optimization problem as a function of the shrinkage parameter is presented.
- A computationally fast filter input selection strategy for the MLP network is proposed. The input variables are selected based on the linear model using a novel backward input elimination strategy. Subsequently, the obtained inputs are used in the MLP network. The applicability of the method is analyzed in detail in the context of the long-term prediction of time series.
- A constraint optimization problem to train the RBF network in the function approximation problem is proposed. The resulting network can be sparse in terms of both input variables and basis functions. In addition, a sequential backward input variable elimination algorithm for the RBF network is presented. Pruning of the inputs is based on a new relevance measure that is determined using partial derivatives of the network.

1.3 Publications of the thesis

The present thesis consists of an introductory part and the following seven peer-reviewed original publications.

1. Timo Similä and Jarkko Tikka (2005). Multiresponse sparse regression with application to multidimensional scaling, *in* W. Duch, J. Kacprzyk, E. Oja, and S. Zadrozny (eds), *Artificial Neural Networks: Formal Models and Their Applications – ICANN 2005, 15th International Conference, Proceedings,*

Part II, Vol. 3967 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 97–102.

2. Timo Similä and Jarkko Tikka (2006). Common subset selection of inputs in multiresponse regression, *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN 2006)*, pp. 1908–1915.

3. Timo Similä and Jarkko Tikka (2007). Input selection and shrinkage in multiresponse linear regression, *Computational Statistics & Data Analysis* **52**(1): 406–422.

4. Jarkko Tikka and Jaakko Hollmén (2008). Sequential input selection algorithm for long-term prediction of time series, *Neurocomputing*, **71**(13–15): 2604–2615.

5. Jarkko Tikka and Jaakko Hollmén (2008). Selection of important input variables for RBF network using partial derivatives, *in* M. Verleysen (ed.) *Proceedings of the 16th European Symposium on Artificial Neural Networks (ESANN 2008)*, d-side publications, pp. 167–172.

6. Jarkko Tikka (2007). Input selection for radial basis function networks by constrained optimization, *in* J. Marques de Sá, L. A. Alexandre, W. Duch, and D. Mandic (eds), *Artificial Neural Networks – ICANN 2007, 17th International Conference, Proceedings, Part I*, Vol. 4668 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 239–248.

7. Jarkko Tikka (2008). Simultaneous input variable and basis function selection for RBF networks, *Neurocomputing*, Accepted for publication.

In the text, the publications are referred using the above numbering, for instance Tikka (2008) is referred to as Publication 7.

1.4 Contents of the publications and author's contributions

Publication 1. The multiresponse sparse regression (MRSR) algorithm for linear regression is proposed. It extends the LARS algorithm by Efron et al. (2004) to the case of multiple responses. The MRSR algorithm updates the model with less greedy steps than a classical forward stepwise input selection strategy, which often improves the prediction performance. Similä suggested the idea originally, but the algorithm itself was jointly developed by the author and Similä. The author carried out the experiments, while the application to multidimensional scaling was outlined by Similä. The article was written together.

Publication 2. The MRSR algorithm is presented in a more general framework. The correlation criterion for model construction is defined using the L_2- and L_∞-norms in addition to the L_1-norm, that is applied in Publication 1. Computational improvements are also introduced. It is additionally shown that the outcome of the algorithm is unique, assuming implicitly that the selected inputs are linearly independent. Extensive experimental comparisons illustrate the strengths of the MRSR algorithm in the case of highly correlated inputs. The MRSR algorithm is further applied to unsupervised input selection in an application of image reconstruction. Similä was more responsible for the theoretical developments, including the proof of uniqueness, but the research was continuously supported by the author. The image reconstruction application was Similä's idea. Otherwise, the experiments were designed, carried out, and reported by the author. Similä wrote the other parts of the article.

Publication 3. The problem of input selection in multiresponse linear regression is formulated as a convex optimization problem. The sum of squared errors is minimized subject to a constraint that encourages sparsity in terms of the inputs. The proposed problem can be seen as an extension of the LASSO formulation (Tibshirani, 1996) to the case of multiple responses. Necessary and sufficient conditions for optimality and a condition that ensures the uniqueness of the solution are given. An interior point method is proposed for solving the optimization problem. A predictor-corrector variant of the solver suits for following the entire solution path as a function of the shrinkage parameter. Extensive empirical comparisons are performed. Similä contributed the theory, implemented the algorithm, and wrote a large part of the article. The author provided insight into the theory and helped considerably in the way toward a workable implementation. Furthermore, the author was responsible for planning, conducting, and reporting the experiments.

Publication 4. The sequential input selection algorithm (SISAL) is proposed for choosing a parsimonious set of inputs in the problem of long-term prediction of time series. The SISAL utilizes the backward elimination strategy in a linear prediction model. The removal of inputs is based on a novel relevance measure that is estimated using a resampling procedure. Subsequently, a nonlinear predictor is constructed applying the inputs that are selected by the SISAL. Importance of the inputs in the nonlinear model is assessed using partial derivatives. Experiments on problems with varying nonlinearity show that importance of the inputs based on the linear model reflect very nicely importance of the inputs in the nonlinear model as well. The SISAL was developed by the author. Experiments were jointly designed with Hollmén and all of them were conducted by the author. The author wrote most of the article.

Publication 5. The sequential input selection algorithm for the RBF (SISAL-RBF) network is presented. One input is deleted at a time starting from the network including all the inputs. Thus, the number of candidate subsets equals to the number of available input variables. The pruning of inputs is based on the ranking of inputs that is evaluated using partial derivatives of the network. Furthermore, one hyperparameter can be tuned automatically due to the efficient determination of the leave-one-out cross validation error. Experiments prove the usefulness of the proposed strategy. The method was developed by the author. The author designed most of the experiments, conducted all of them, and wrote most of the article.

Publications 6 and **7**. Input variable selection for the RBF network is formulated as a constrained optimization problem. Complexity of the network is controlled by two hyperparameters. Each input dimension is weighted by a continuous valued weighting factor. The first hyperparameter constrains the L_1-norm of the factors which encourages sparsity in terms of the inputs. The optimization of the weighting factors is based on the data and the resulting weights reflect relevance of the input variables. The second hyperparameter is related to the output layer parameters of the network. The model fitting is penalized by the L_2-norm of the parameters in Publication 6, whereas it is controlled by the constraint implemented by the L_1-norm of the parameters in Publication 7. The advantage of the L_1-norm over the L_2-norm is that sparsity is also achieved in terms of the basis functions. In Publication 6, the problem is solved using an alternating optimization approach. In Publication 7, a direct optimization procedure is also presented and compared to an alternating optimization strategy. Convincing results from extensive experiments and comparisons on both simulated and real world data sets justify the advantages of the proposed approach. The present author was the sole contributer to these articles.

1.5 Structure of the thesis

The rest of the introductory part is organized as follows. Chapter 2 briefly reviews different learning tasks in data analysis and presents the framework of the present thesis. The problem of variable selection is also motivated and introduced in detail. Chapter 3 begins with a discussion on variable selection for the single response linear regression model followed by extensions to the multiresponse case. Chapter 4 includes input variable selection strategies for nonlinear regression models. It begins by discussing the learning of an unknown nonlinear function by nonadaptive methods and support vector machines. Chapter 4 continues by a short introduction to artificial neural networks and a more detailed description of variable selection methods for neural networks. Chapters 3 and 4 contain examples that illustrate advances of the proposed methods. Finally, a summary of the thesis with some conclusions is presented in Chapter 5.

Chapter 2

Data analysis

2.1 Overview

The purpose of data analysis is to extract useful information from a large collection of data. Traditionally this has been manual work. Due to the dramatic increase in sizes of data sets, knowledge extraction needs to be automated. However, a fully automatized system cannot be expected to work properly since a domain expert is required, for instance, to judge the relevance of the findings (Mannila, 1997).

Figure 2.1 illustrates the main parts of a data analysis process (Tukey, 1980; MacKay and Oldford, 2000). The further phases are dependent on the previous ones, i.e choices that are done affect the subsequent stages. However, it is also ordinary to go back to the previous steps, although it is not shown in the figure. During the data analysis process new knowledge is typically obtained and the previous choices may have to be reformulated (Feelders et al., 2000).

The problem statement defines what is to be learned and it also affects the design of the following phases. A classical task is to estimate a model to describe dependencies between input variables and a response variable with two goals (Breiman, 2001). First, the model should be able to predict the response accurately based on the input variables. The second objective is to assist in interpretation of importance of the inputs in the prediction task. Methods that are proposed in the present thesis are designed to fulfill these two objectives.

The study plan specifies the collection of data. It includes the selection of the response and inputs variables. The response should describe some feature of a system or phenomenon as clearly as possible, such that it enables the discovery of novel knowledge. All the possible variables that may be useful in the prediction of the response should also be detected. The collection of data includes measuring the chosen variables. In experiments of the present thesis, it was not possible to affect the design of data sets except in the cases of simulated data.

Figure 2.2 shows a diagram of the model construction. Training and evaluation of the model should be based on separate data sets that are independent from each

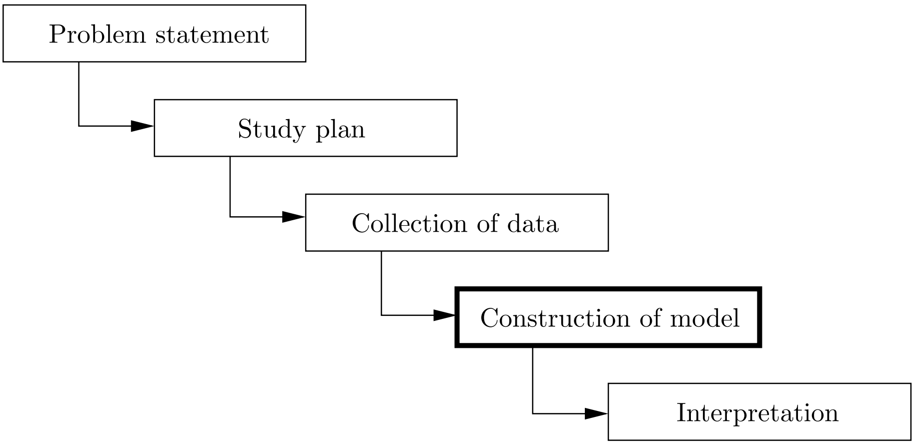

Figure 2.1: The main steps of a data analysis process. The present thesis concentrates on the construction of model.

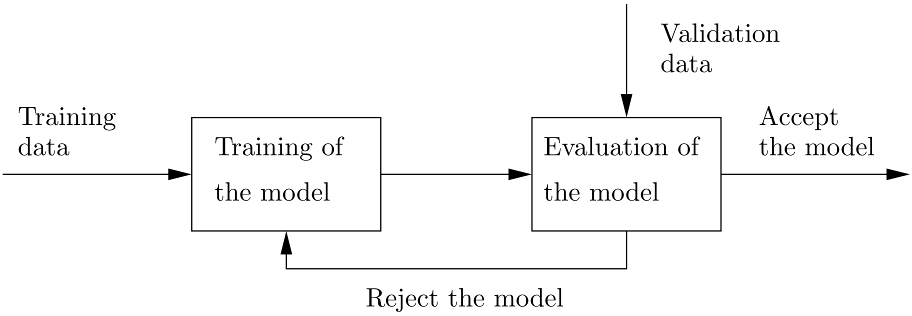

Figure 2.2: A flow chart of the model construction.

other (Bishop, 1995; Hand et al., 2001). The model is constructed using a training data set. If the model performs sufficiently well on validation data the model is accepted to further analyses. Otherwise, the model is retrained with revised initial conditions. The main contributions of the present thesis contain input variable selection methods for the construction of linear and nonlinear regression models.

In the interpretation step, the final model is used to answer the questions proposed in the problem statement. The iterations between the training and validation sets can introduce overfitting, that is, a model approximates training data accurately but it performs poorly on novel data. It indicates that the model fits the noise and it is overly complex. On the other hand, an insufficiently complex model is not able to predict either training or novel observations. Thus, the prediction performance of the final model should be assessed using the third independent data set known as a test data set. Finally, the model should be interpreted, for instance, the inputs that explain the most of the variation in the response should be indicated. Input variable selection facilitates interpretation, since useless inputs are not included into the final model at all.

2.2 Learning tasks in the data analysis

2.2.1 General framework

Exploratory data analysis (EDA) is usually the first step in the examination of multivariate data (Tukey, 1977). EDA consists of graphical and numerical techniques to examine data, for instance, box plots, scatterplots, and principal component analysis, correlation analysis, and factor analysis (Everitt and Dunn, 1991). In some cases, EDA may be sufficient for summarizing a data set. On the other hand, it may reveal that the quality of the data is not good enough for inference of complex models. EDA is also advantageous in hypotheses generation for confirmatory data analysis as noted by Tukey (1980) and Chatfield (1995).

Confirmatory data analysis (CDA) can be applied when statistical hypotheses are available for the problem under study. Statistical testing requires the null and alternative hypotheses that are mutually exclusive. The null hypothesis proposes a statement, which is assumed to be true. It is rejected, i.e the alternative hypothesis is accepted, when observations contain enough information to conclude that it is not probable to obtain such result by a chance under the null hypothesis. Milton and Arnold (1990) present a wide variety of statistical tests for various applications. The parametric models and strong assumptions limits the applicability of statistical tests in complex problems (Breiman, 2001).

In the algorithmic data analysis approach, it is typically assumed that the data contain independently and identically distributed samples from an unknown probability distribution (Breiman, 2001). Learning from a finite number of samples by modern algorithmic methods is also known as machine learning (Cherkassky and Mulier, 1998). In the machine learning methods, the focus is traditionally laid on prediction accuracy of the models and training relies heavily on optimization strategies. In unsupervised learning, desired outputs of the model are unknown and the model is adapted to a data set to represent its internal structure by minimizing a representation error criterion (Hinton and Sejnowski, 1999; Oja, 2002). For example, semiparametric density estimation, feature extraction, and clustering are unsupervised learning tasks. In supervised learning, the desired responses for the inputs are known (Haykin, 1999). The mapping between the inputs and the model output is determined by minimizing an error function between the model output and the desired responses. Classification and regression problems are examples of supervised learning tasks.

2.2.2 Density estimation

Let $\boldsymbol{x}_n, n = 1, \ldots, N$ denote independently and identically distributed samples from an unknown probability density function $p(\boldsymbol{x})$. The task is to estimate that function from the samples $\boldsymbol{x}_n$ with subject to modelling constraints.

A classical approach is to specify a parametric density model $p(\boldsymbol{x}, \boldsymbol{\omega})$. The adjustable parameters $\boldsymbol{\omega}$ are chosen by maximizing the log-likelihood function of the

N observations $\boldsymbol{x}_n$

$$\underset{\boldsymbol{\omega}}{\text{maximize}} \quad \mathcal{L} = \sum_{n=1}^{N} \log \{p(\boldsymbol{x}_n, \boldsymbol{\omega})\} \ . \tag{2.1}$$

Cherkassky and Mulier (1998) state that according to the maximum likelihood inductive principle, the observed data have been generated most likely by the distribution model $p(\boldsymbol{x}, \boldsymbol{\omega}^*)$, where $\boldsymbol{\omega}^*$ is the optimal solution for the problem in Equation (2.1). In some cases, a parametric density model may be overly restricted, since a model with a finite number of parameters cannot represent all distributions.

An alternative is to use nonparametric approaches, that typically can model any distribution if there is enough data. However, the computational complexity grows with increasing amount of data, which may make nonparametric models impractical (Bishop, 1995). Moreover, the number samples that is required for the accurate estimation increases rapidly as the dimensionality of a problem grows (Cherkassky and Mulier, 1998). One of the simplest models is a histogram, but it is only applicable for low-dimensional data. In the kernel density estimation, a kernel function $K(\boldsymbol{x}, \boldsymbol{x}_n)$ is located on each data point $\boldsymbol{x}_n$ and the density is estimated as a sum of the kernel functions

$$p(\boldsymbol{x}) = \sum_{n=1}^{N} K(\boldsymbol{x}, \boldsymbol{x}_n) \ . \tag{2.2}$$

A hypercube kernel is known as a Parzen window (Parzen, 1962), but a smoother density function is obtained, for instance, by a multivariate normal kernel function (Bishop, 1995).

In a finite mixture model, complexity and flexibility of a density model can be controlled systematically (McLachlan and Peel, 2000). A finite mixture model is a linear combination of parametric density functions

$$p(\boldsymbol{x}) = \sum_{j=1}^{J} \pi_j p(\boldsymbol{x}|j) \ , \tag{2.3}$$

where π_j are mixture proportions with the properties $\pi_j \geq 0$ and $\sum_{j=1}^{J} \pi_j = 1$. The log-likelihood for a data set is

$$\mathcal{L} = \sum_{n=1}^{N} \log \left[\sum_{j=1}^{J} \pi_j p(\boldsymbol{x}_n|j) \right] . \tag{2.4}$$

The maximum likelihood estimates can be calculated by the Expectation-Maximization algorithm (Dempster et al., 1977; Redner and Walker, 1984). The choice of the component distribution $p(\boldsymbol{x}|j)$ depends on the problem. For instance, Tikka et al. (2007) and Myllykangas et al. (2008) model DNA copy number amplification data using finite mixtures of multivariate Bernoulli distributions.

2.2.3 Feature extraction

In a feature extraction problem, the objective is to transform data from a high-dimensional space into a low-dimensional one. Dimensionality is decreased by forming linear or nonlinear combinations of the original variables. The combinations, known as features, should preserve as much information as possible from the original data. The number of features is typically clearly smaller than the number of original variables which speeds up subsequent analyses.

Let us assume that each dimension of samples $\boldsymbol{x}_n^T = [x_{n1}, \ldots, x_{nd}]$, $n = 1, \ldots, N$ has zero mean and unit variance. In principal component analysis (PCA), a following linear mapping

$$\boldsymbol{z} = \boldsymbol{Q}^T \boldsymbol{x} \tag{2.5}$$

is sought, where $\boldsymbol{z}^T = [z_1, \ldots, z_k]$ and $k < d$ (Haykin, 1999). To obtain an orthogonal transformation that preserves the maximum amount of variance of the original data, the columns of the matrix $\boldsymbol{Q}$ have to be the eigenvectors of the sample covariance matrix of the observed vectors $\boldsymbol{x}_n$. The k eigenvectors, i.e. the loadings of the principal components, are selected according to the k largest eigenvalues.

Independent component analysis (ICA) can be used to decompose a data set into independent parts (Hyvärinen and Oja, 2000; Hyvärinen et al., 2001). ICA is based on a latent variable model

$$\boldsymbol{x} = \boldsymbol{A}\boldsymbol{s} \ , \tag{2.6}$$

in which the observed variables $\boldsymbol{x}$ are considered mixtures of independent components or sources $\boldsymbol{s}$. Both sources $\boldsymbol{s}$ and mixing matrix $\boldsymbol{A}$ are unknown and they have to be estimated. The sources $\boldsymbol{s}$ are assumed to be statistically independent and distributed according to non-Gaussian distributions. In addition, to simplify estimation it is usually assumed that the matrix $\boldsymbol{A}$ is square, i.e the number of sources equals to the number of variables. A fast and statistically robust algorithm to find the independent components is presented by Hyvärinen (1999).

Nonlinear versions of both PCA and ICA have also been developed. Schölkopf et al. (1998) present a kernel PCA algorithm to evaluate the standard PCA in a high-dimensional feature space. Calculations are done implicitly in that space using kernel functions and the resulting principal components and the original inputs are nonlinearly dependent. With large data sets, the kernel PCA is computationally unattractive since a kernel function is placed on each data point. However, Tipping (2001) presents an extension based on a maximum-likelihood approach, that uses only a relatively small subset of the training samples. A natural extension of the ICA model in Equation (2.6) is a nonlinear mixing model

$$\boldsymbol{x} = \boldsymbol{F}(\boldsymbol{s}) \ , \tag{2.7}$$

where $\boldsymbol{F}$ is a real and vector valued mixing function (Jutten and Karhunen, 2004). In general, the number of solutions is infinite but an unique solution can be obtained by imposing constraints in the estimation (Hyvärinen and Pajunen, 1999).

Autoassociative neural networks can also be applied to linear (Baldi and Hornik, 1989) and nonlinear (Kramer, 1991; Oja, 1991) feature extraction. The same data is used as the inputs and the responses of the network. Extracted features are the outputs of the hidden neurons and the amount of compression is controlled by the number of neurons. In the case of linear autoassociative network, the minimization of mean squared error has the unique global optimum and the solution is equivalent with PCA (Baldi and Hornik, 1989, 1995). Japkowicz et al. (2000) demonstrate that nonlinearities in the hidden layer are indeed required in some applications and resulting features are not equal to the principal components. The selection of a proper number of neurons is however more problematic in the nonlinear case. DeMers and Cottrell (1993) propose to observe the variances of representation units and to prune an unit if its variance is below a predefined threshold. Hsieh (2007) suggests to control complexity of the network by detecting neighboring samples that are projected far apart from each other.

According to Carroll and Arabie (1980), any nonlinear mapping technique that searches a low-dimensional representation of high-dimensional data based on given proximities is known as multidimensional scaling. The representation aims to preserve similarities d_{ij} between features $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$ as close as possible to given similarities δ_{ij} between the corresponding original samples $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. A two- or three-dimensional feature space is commonly used for reasons of visualization. If no a priori knowledge is available, similarities are typically measured by the Euclidean distance. The following well-known and widely applied criterion

$$\underset{d_{ij}}{\text{minimize}} \quad E = \frac{1}{\sum_{i<j}^{N} \delta_{ij}} \sum_{i<j}^{N} \frac{(d_{ij} - \delta_{ij})^2}{\delta_{ij}} \tag{2.8}$$

to rank different representations is proposed by Sammon (1969). A drawback of solutions of the problem in Equation (2.8) is that only the training samples can be mapped to the feature space. Webb (1995) and Lowe and Tipping (1996) consider a parameterized transformation $\boldsymbol{z} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{w})$ which allows also mapping of other samples. They implement the function $\boldsymbol{f}$ by a RBF network and optimize a representation criteria directly as a function of the network parameters. Tipping and Lowe (1998) propose the computationally more efficient shadow targets algorithm. In Publication 1, the linear mapping $\boldsymbol{z} = \boldsymbol{W}\boldsymbol{x}$ is utilized instead of the nonlinear one and MRSR and the shadow targets algorithm are combined, which results in a subset of the original variables that are used in the feature extraction.

The main disadvantage of most feature extraction methods is that extracted features are combinations of all the original input variables. The dimensionality of the problem is decreased, but the interpretation of the features may still be difficult. Even relative importance of the original inputs is commonly unclear.

2.2.4 Clustering

Cluster analysis is ordering of the observed samples $\boldsymbol{x}_n, n = 1, \ldots, N$ into natural and meaningful groups or clusters (Jain et al., 1999; Xu and Wunsch, 2005). In a valid clustering, the samples belonging to the same cluster are more similar to each other than they are to the samples belonging to the other clusters. Prototypes of the clusters provide a compact representation of the data (Hollmén and Tikka, 2007).

Since the clustering is based on similarities and dissimilarities of the samples, the selection of a proximity measure greatly affects results. The most common strategy is perhaps to evaluate dissimilarities between the samples using some distance measure. In the case of continuous valued inputs, for instance the City-block, Euclidean, Minkowski, or Mahalanobis distance can be applied. Ichino and Yaguchi (1994) consider the case including both quantitative and qualitative variables.

Hierarchical and partitional algorithms are perhaps the most used methods in cluster analysis (Hansen and Jaumard, 1997). A hierarchical algorithm produces a nested series of partitions and represents the data as a dendrogram. The root node corresponds to the whole data set and each leaf node represents a single sample. The final clusterings are obtained by cutting the dendrogram at intermediate levels. Most hierarchical clustering algorithms are variants of the single- or complete-linkage algorithms (Jain et al., 1999). A partitional method produces a single clustering of the data instead of a clustering structure. The k-means (Bishop, 1995) and support vector clustering (Ben-Hur et al., 2001) algorithms are examples of the partitional methods. Partitional algorithms are not as flexible as hierarchical ones since the number of clusters needs to be predefined. On the other hand, it is faster to evaluate a single partition than a dendrogram especially in the case of large training data.

The assessment of the clustering results is crucial, because each clustering algorithm produces a partition of the data whether the data is separable or not. The external validation is based on a priori knowledge of the data, which is, however, rarely available. The ranking of different clusterings according to indices, that are evaluated from the data, is known as internal validation. Bezdek and Pal (1998); Maulik and Bandyopadhyay (2002) discuss and compare several indices for internal validation.

2.2.5 Classification

In a classification problem, a discrete class label $C_k, k = 1, \dots, K$ of each input vector $\boldsymbol{x}_n$, $n = 1, \dots, N$ is known. The objective is to to assign novel input vectors to one of the given classes.

Bishop (1995) shows that the probability of misclassification of a novel sample $\boldsymbol{x}$ is minimized if it is assigned to the class C_k having the largest posterior probability $P(C_k|\boldsymbol{x})$. The posterior probabilities are evaluated using Bayes' theorem

$$P(C_k|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|C_k)P(C_k)}{p(\boldsymbol{x})} \quad , \tag{2.9}$$

where $p(\boldsymbol{x}|C_k)$ is the class conditional likelihood of the sample $\boldsymbol{x}$ and $P(C_k)$ is a priori probability of the class C_k. The unconditional density $p(\boldsymbol{x})$ is independent of the classes, thus the sample $\boldsymbol{x}$ is assigned to the class C_k if $p(\boldsymbol{x}|C_k)P(C_k) > p(\boldsymbol{x}|C_j)P(C_j)$ for all $j \neq k$. The class conditional likelihoods can be estimated, for instance, using the methods presented in Section 2.2.2.

The classification process can be reformulated by discriminant functions $y_k(\boldsymbol{x})$. A natural choice is $y_k(\boldsymbol{x}) = p(\boldsymbol{x}|C_k)P(C_k)$, but other functions can be applied as well. In general, on decision boundaries the discriminant functions are equal

$y_k(\boldsymbol{x}) = y_j(\boldsymbol{x})$. In a two class ($K = 2$) problem, only one discriminant function is required. It is defined such that $y(\boldsymbol{x}) > 0$ if $\boldsymbol{x}$ is from the class C_1 and $y(\boldsymbol{x}) < 0$ if $\boldsymbol{x}$ is from the class C_2. In the rest of the section, only the two class case is considered.

A classical discriminant function is a linear combination of the input variables $y(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x}$. Fisher (1936) presents that the best separating function is obtained by maximizing the ratio of between-class to within-class measures, which are defined by the distance between projected means and the sum of variations of projected samples within classes, respectively. The assumption of linearity is restrictive for many applications. Thus, Mika et al. (1999) propose to evaluate Fisher's linear discriminant implicitly in a feature space using kernel functions resulting into a nonlinear discriminant function in the original input space.

In the case of linearly separable classes, a separating hyperplane calculated by the support vector machine (SVM) is closely related to Fisher's discriminant function as shown by Shashua (1999). The separating hyperplane is defined such that the margin between the classes is maximized. In the linearly inseparable case, the separating hyperplane can be evaluated in a high-dimensional feature space (Cristianini and Shawe-Taylor, 2000). It is highly probable that the data is linearly separable in a feature space if the transformation is nonlinear and dimensionality of the feature space is high enough (Haykin, 1999). No explicit definition of the feature space is typically needed, since calculations are done implicitly by kernel functions, that allows even infinite dimensional feature spaces (Burges, 1998).

In a linear logistic regression model (Myers and Montgomery, 1997), the logarithm of the ratio of the posterior probabilities is assumed to be linear

$$y(\boldsymbol{x}) = \log \frac{P(C_1|\boldsymbol{x})}{P(C_2|\boldsymbol{x})} = w_0 + \sum_{i=1}^{d} w_i x_i = \boldsymbol{w}^T\boldsymbol{x} \ . \tag{2.10}$$

The bias w_0 is included to the parameter vector $\boldsymbol{w}$ by adding a constant variable to the input vector $\boldsymbol{x}$. The posterior probabilities can be further written as

$$P(C_1|\boldsymbol{x}) = \frac{e^{\boldsymbol{w}^T\boldsymbol{x}}}{1 + e^{\boldsymbol{w}^T\boldsymbol{x}}} \quad \text{and} \quad P(C_2|\boldsymbol{x}) = 1 - P(C_1|\boldsymbol{x}) = \frac{1}{1 + e^{\boldsymbol{w}^T\boldsymbol{x}}} \ . \tag{2.11}$$

The maximum likelihood estimates of the parameters $\boldsymbol{w}$ are evaluated by minimizing the negative conditional log-likelihood of the classes using the observed samples

$$\mathcal{L} = -\sum_{n=1}^{N} y_n \log P(C_1|\boldsymbol{x}_n) + (1 - y_n)P(C_2|\boldsymbol{x}_n) \ , \tag{2.12}$$

where $y_n = 1$ if the observation $\boldsymbol{x}_n$ belongs to the class C_1 and $y_n = 0$ if $\boldsymbol{x}_n$ belongs to the class C_2. A nonlinear decision boundary is obtained by setting $y(\boldsymbol{x}) = f(\boldsymbol{x}, \boldsymbol{w})$ and learning the function f, for example, using neural networks (Schumacher et al., 1996; Vach et al., 1996) or kernel functions (Zhu and Hastie, 2005). Regardless of the functional form of the decision boundary, iterative optimization methods are required in the determination of the parameters $\boldsymbol{w}$.

2.2.6 Regression

In a regression problem, the task is to find a functional representation between an output y and input $\boldsymbol{x} = [x_1, \ldots, x_d]$ variables. The output variable is formulated as the sum of a deterministic function f and a random error ε as follows

$$y = f(\boldsymbol{x}, \boldsymbol{w}) + \varepsilon \ , \tag{2.13}$$

where $\boldsymbol{w}$ is a vector of adjustable parameters. The estimation of the function f is based on a finite set of observations $\mathcal{D} = \{y_n, \boldsymbol{x}\}_{n=1}^N$. It is typically assumed that the additive errors ε_n are independently and identically distributed.

Bishop (1995) shows that if the errors are independently and normally distributed with zero mean and a common finite variance $\varepsilon_n \sim \mathrm{N}(0, \sigma^2)$, the maximization of the likelihood of the errors $y_n - f(\boldsymbol{x}_n, \boldsymbol{w})$ is equivalent to the minimization of the sum of squared errors

$$\mathrm{SSE} = \sum_{n=1}^{N} (y_n - f(\boldsymbol{x}_n, \boldsymbol{w}))^2 \ . \tag{2.14}$$

In regression problems, SSE is perhaps the most commonly used cost function and it is minimized with respect to the model parameters $\boldsymbol{w}$. The model output is given by the conditional average of the output data if the following three requirements are satisfied (Bishop, 1995). First, the number of observations N is large enough to approximate an infinite data set. Second, the function f is sufficiently general. Third, an appropriate minimum for the cost function is found.

A time series prediction problem is an example of the regression tasks. A time series is an ordered sequence of observations from a variable y at time t. The common ordering is through equally spaced time intervals denoted by y_t, $t = 1, \ldots, T$, but it can also be taken through other dimensions, such as space (Wei, 2006). The objective is to predict future values of the time series y_{t+h}, $h = 0, 1, 2, \ldots$ using the model

$$\hat{y}_{t+h} = f(y_{t-1}, \ldots, y_{t-d_1}, \boldsymbol{x}_{t-1}, \ldots, \boldsymbol{x}_{t-d_2}, \boldsymbol{w}) \ , \tag{2.15}$$

where $y_{t-i}, i = 1, \ldots, d_1$ are the past values of the time series itself and $\boldsymbol{x}_{t-j}, j = 1, \ldots, d_2$ are the past values of exogenous variables (Chatfield, 2001). Parameters $\boldsymbol{w}$ are optimized by minimizing some cost function, usually the sum of squared errors, between the observations y_{t+h} and the model outputs $\hat{y}_{t+h}$.

Simple linear functions $f(\boldsymbol{x}, \boldsymbol{w}) = \boldsymbol{w}^T \boldsymbol{x}$ are often considered uninteresting, especially in complex real world applications, due to limited modeling capacity. However, several reasons support the usage of the linear models (Baldi and Hornik, 1995; Hand et al., 2001). The estimation of the parameters is straightforward and computationally fast. The resulting models can be analyzed using well-defined statistical tests (Draper and Smith, 1981), that cannot be generally derived in the cases of nonlinear functions. Interpretation of the models is uncomplicated, because estimated values of the parameters describe directly contributions of the inputs in the model. Linear models can also be used as a baseline for more complex models. Linear regression is discussed further in Chapter 3.

In some cases, it is known that the input-output relationship is nonlinear, but the exact functional form is still unknown. Support vector machines (Smola and

Schölkopf, 2004) and artificial neural networks (Haykin, 1999) are flexible and powerful models for the learning of nonlinear functions. It is shown by Hornik et al. (1989, 1990); Hartman et al. (1990); Park and Sandberg (1991, 1993) that a feedforward neural network with only one hidden layer is able to approximate arbitrarily well any finite dimensional smooth continuous mapping if the number of neurons in the hidden layer is sufficiently large. However, flexibility makes nonlinear regression models prone to overfitting (White, 1990; Zhang, 2007). General strategies to decrease possibility of overfitting are discussed in the next section, whereas various techniques to learn a nonlinear regression function are presented in Chapter 4.

2.3 Model complexity

As an example, let us consider a data set that includes $N = 50$ samples from an input x and an output y. The input samples are generated independently from a uniform distribution $\mathcal{U}(-2, 3)$. The noisy output samples $y = t + \varepsilon$ are obtained by adding independently normally distributed errors $\varepsilon \sim \mathrm{N}(0, 2^2)$ to the values of the target function

$$t(x) = 1 - 4x - x^2 + x^3 \quad . \tag{2.16}$$

Figure 2.3 illustrates three estimates of the target function. A linear model does not perform adequately in this task since it is not flexible enough, see Fig. 2.3(a). An approximation that is obtained by an over-complex neural network is shown in Fig. 2.3(b). The over-complex network follows the data points but oscillates heavily. A smooth approximation by a network with proper complexity is illustrated in Fig. 2.3(c). All the three approximations and the actual target function $t(x)$ are presented in Fig. 2.3(d). However, visual justification is not possible in high-dimensional problems. Therefore, data-based techniques are used to define complexity in practice.

The trade-off between model complexity and prediction accuracy is referred as the bias-variance dilemma (Geman et al., 1992). A simple model has a large bias, which means that the model output is on average different from the underlying true function. Sensitivity of the model output to various data sets is called variance. Since the bias and variance are complementary measures, a compromise between them is required in order to achieve the optimal generalization performance. Basically, a complexity of a regression model depends on the number of tunable parameters. For instance, in a neural network the number of parameters can be varied by the numbers of hidden neurons and input variables (Reed, 1993). Alternatively, the number of effective parameters can be decreased by regularization techniques (Girosi et al., 1995).

In the ideal case, three separate data sets are available for the model construction. Models with varying complexity are estimated using training data by optimizing a fitting criterion. The trained models are evaluated using validation data, and the model that maximizes a performance criterion is selected to further analyses. The prediction performance of the selected model is finally assessed using test data. Comparisons between different methods should be performed using the test data set, since it is not applied in the training and model selection phases at all.

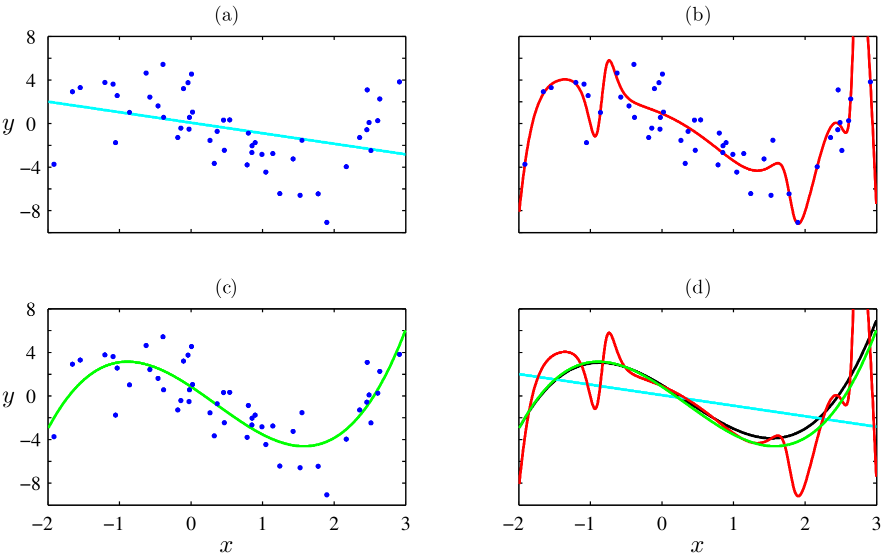

Figure 2.3: Various regression functions that are learnt based on the data points (*the blue dots*). (a) An inflexible model. (b) An over-flexible model. (c) A smooth approximation. (d) All the above approximations and the correct function (*the black line*).

However, the number of data points is typically limited. Hence, it is not possible to have a separate representative validation data set. In such a case, cross-validation (CV) is a generally applicable model selection strategy (Bishop, 1995). In k-fold CV, the data is divided into k separate parts. The model is trained using $k-1$ parts and the remaining kth part is used for the validation. The process is repeated k times such that the validation is carried out using each of the k parts in turn. The overall validation error is the average of these k results. In the extreme case called leave-one-out CV, only one sample is left out from the training in turn, which makes it closely related to the jackknife (Efron, 1982). Shao (1993) shows that the leave-one-out CV is asymptotically inconsistent in the case of linear models, i.e the probability of selecting the model with the best predictive ability does not converge to unity as the number of observations approaches infinity. However, leave-one-out CV is often applied with linear models, since an estimate of the generalization error can be calculated in a closed form without the repetitive training process (Orr, 1996). The bootstrap (Efron and Tibshirani, 1993) is also based on the resampling of the original data and it can be applied to model selection as well. Kohavi (1995) performs extensive experiments to compare CV with the bootstrap and concludes to recommend 10-fold CV based on the results.

Instead of resampling techniques, the model complexity can be appropriately selected using an information criterion. Typically, it is defined as the sum of two components. The first term measures the accuracy of the model on the training data and the second one penalizes from the complexity of the model, i.e no separate validation data is needed. The optimal value for an information criterion results into a compromise between accuracy and smoothness of the model. Information criteria are proposed for various cases, for instance, for a linearly parameterized model (Mallows, 2000), for an unregularized model trained by the maximum likelihood principle (Akaike, 1974), and for a regularized nonlinear system (Moody, 1991, 1992; Larsen and Hansen, 1994).

2.4 Variable selection

Lately, variable selection methods have gained more and more attention in several data analysis problems, especially in the applications that contain hundreds or thousands of variables. Guyon and Elisseeff (2003) motivate the usage of variable selection algorithms by the following three arguments.

1. The prediction performance of the constructed model may be improved by discarding irrelevant variables.
2. A more cost-effective model is obtained, that is, training of the model can be faster, and measuring and storage requirements of the data are decreased.
3. The model provides better understanding of the system that has generated the data, since the model includes only a subset of available variables.

Although the number of variables would be large, the number of samples is limited in many cases. However, the number of data points that is required to estimate the underlying phenomenon accurately increases exponentially as a function of the dimension of data, which is known as the curse of dimensionality. Thus, fitting a complex model using an insufficiently small number of high-dimensional data points probably leads to overfitting. Verleysen (2003) discusses difficulties related to high-dimensional data and presents examples of surprising distributions of high-dimensional data points. Feature extraction and variable selection methods can be used to circumvent problems that are encountered in high-dimensional spaces.

2.4.1 Relevant variables

Blum and Langley (1997); Kohavi and John (1997) present several definitions of relevance of a variable, which are applied in different contexts. In general, weak and strong degrees of relevance are needed (John et al., 1994). A strongly relevant variable x_i is indispensable for the model, i.e. the removal of x_i from the model deteriorates the prediction performance. The strong relevance means that the response y and an input x_i are statistically dependent. A weakly relevant variable is not strongly relevant and it can sometimes contribute to performance of the model. An input x_i is weakly relevant if there exists a subset of the inputs $\mathcal{S}$, $x_i \notin \mathcal{S}$, such that y and x_i are conditionally dependent given $\mathcal{S}$. An input variable is irrelevant if it is neither strongly nor weakly relevant.

Instead of the estimation of statistical dependencies, the present thesis focuses on constructing accurate regression models using a parsimonious subset of the original input variables. The objective is to find a subset of useful variables:

Definition. Let input variables $\boldsymbol{x} = [x_1, \ldots, x_d]$ and a model $\mathcal{M}$ be given. A useful subset of the original inputs $\boldsymbol{x}_u \subseteq \boldsymbol{x}$ maximizes the generalization performance of the model $\mathcal{M}$.

Guyon and Elisseeff (2003) point out that the subset of useful variables does not necessarily contain all the relevant variables. Redundant, but relevant, variables may be especially excluded from the subset of useful variables. Thus, finding the useful subset contradicts finding all the relevant variables. In addition, Blum and Langley (1997) give an example, that a variable can contain relevant information but it is useless from the prediction point of view. Irrelevant variables are rarely useful, and they should be removed from the model (Kohavi and John, 1997). In the rest of the thesis, a relevant, informative, and important variable refer to the useful variable.

The optimal useful subset of variables could be selected by evaluating all the possible subsets. The exhaustive search is computationally infeasible in high-dimensional problems, since the number of subsets increases exponentially as the number of available inputs increases. Therefore, computationally more efficient input selection strategies are required to approximate the exhaustive search.

2.4.2 Filter approach

In a filter approach, input variable selection and learning of the final prediction model are independent from each other (Blum and Langley, 1997). First, input selection is performed using a simple model, such as, a linear function (Bi et al., 2003) or a polynomial (Rivals and Personnaz, 2003; Li and Peng, 2007). Another alternative is to use mutual information (François et al., 2007), which is computationally more complex than linearly parameterized models, but it allows to detect more general dependencies between the input and output variables. Second, a nonlinear prediction model using the selected inputs is trained. Since input selection is a preprocessing step, any prediction method can be applied. Filter approaches make a compromise between computational complexity and prediction accuracy. The final nonlinear prediction model needs to be constructed only once, but it is not guaranteed that the selected inputs are useful for the final model. In Publication 4, a filter input selection strategy for a MLP network in the context of long-term prediction of time series is proposed.

2.4.3 Wrapper approach

A wrapper approach is a simple and widely applicable technique to address the problem of input selection (John et al., 1994; Kohavi and John, 1997). A prediction model is considered as a black box and it is retrained with various subsets of inputs. The subsets are ranked according to an estimate of the generalization error and the one having the smallest error is chosen. Forward and backward selection algorithms are classical methods to produce candidate subsets. Forward selection is started from the empty set and an input is added to the current subset in each step of the algorithm. In backward selection, the full set of inputs is reduced by stepwise deletions. Some models are faster to train with only a few variables, but interacting inputs may be detected more easily if selection is started from the full set. Reunanen (2003) finds that greedy search strategies, such as forward selection, are not prone to overfitting. If a criterion for variable selection is monotonic, a branch and bound (BB) algorithm finds the optimal subset of inputs (Narendra and Fugunaga, 1977). However, the computational complexity of the BB algorithm is usually exponential (Kohavi and John, 1997), but Somol et al. (2004) present some heuristics to speed up the algorithm. In the case of a large number of input variables and a complex prediction model, the wrapper approach is computationally unattractive due to the required repeated training of the prediction model.

2.4.4 Embedded methods

In an embedded input selection method, the search of candidate subsets of inputs is guided by approximating changes in the objective function induced by moving in the input space (Guyon and Elisseeff, 2003). Computational requirements are reduced by avoiding retraining of the model for every candidate variable to be eliminated. Guyon et al. (2002) evaluate differences in the objective function of the SVM classifier that are incurred by varying the subset of inputs and keeping other parts of the model fixed. Refenes and Zapranis (1999) rank the input variables using various relevance measures based on the partial derivatives, that are determined with respect to the model output or the error function. Le Cun et al. (1990); Hassibi and Stork (1993) apply the second order Taylor series to approximate changes in the objective function caused by pruning a weight of the MLP network. In Publications 1 and 2, the inputs are added to a multiresponse linear regression model using a correlation criterion between the input variables and the current residuals of the model. In Publication 5, a backward input pruning strategy for the RBF network is proposed. It includes features of both embedded method and wrapper approach, since pruning is based on the partial derivatives and the final subset of inputs is selected using leave-one-out CV.

In a direct embedded method, an objective function consisting of two terms is formalized (Guyon and Elisseeff, 2003). The first term evaluates the goodness-of-fit and the second term measures the number of input variables. In practice, the second term is replaced with a regularization term that restricts complexity of the model. The objective function includes, however, complementary requirements since the optimization of the first term leads to an insufficient solution for the second term, and vice versa. The trade-off between the goodness-of-fit and the amount of regularization is controlled by a hyperparameter such that the generalization performance of the model is maximized. Tibshirani (1996) introduces the least absolute shrinkage and selection operator (LASSO) for a single response linear regression model. In the LASSO formulation, the model fitting is constrained by forcing the sum of the absolute values of the parameters to be less than a predefined constant, which may result into a parsimonious set of inputs. In Publication 3, the LASSO formulation is extended to the case multiple responses. The LASSO type constraints are also applied to the simultaneous shrinkage and selection of the input variables for the RBF network in Publications 6 and 7.

Chapter 3

Variable selection in linear regression

3.1 Single response linear regression

In a linear regression problem, the objective is to estimate a response variable by a linear combination of input variables (Draper and Smith, 1981). Let us consider the case that there are N continuous-valued observations from a one-dimensional response y and d inputs $\boldsymbol{x} = [x_1, \ldots, x_d]$. The standard linear regression model is

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon} \ , \tag{3.1}$$

where $\boldsymbol{y}$ is an $N \times 1$ vector of response values and $\boldsymbol{X}$ is an $N \times d$ matrix of input values. The columns $\boldsymbol{x}_i, i = 1, \ldots, d$ of the matrix $\boldsymbol{X}$ correspond to the variables and the rows $\underline{\boldsymbol{x}}_n, n = 1, \ldots, N$ refer to the observations. The common assumption is that elements ε_n, $n = 1, \ldots, N$ of a noise vector $\boldsymbol{\varepsilon} = [\varepsilon_1, \ldots, \varepsilon_N]^T$ are independently and normally distributed with zero mean and a common finite variance $\varepsilon_n \sim \mathrm{N}(0, \sigma^2)$. The regression coefficients $\boldsymbol{\beta} = [\beta_1, \ldots, \beta_d]^T$ are unknown. The bias term is excluded from the model in Equation (3.1), since it is assumed that all the variables are standardized to have zero mean. Usually, the variables are additionally standardized to have similar scales.

Bishop (1995) shows that under the previous assumptions, the maximum likelihood estimates of the regression coefficients minimize the mean of squared errors (MSE) between the observed and estimated responses

$$\underset{\boldsymbol{\beta}}{\text{minimize}} \quad \frac{1}{N}\|\boldsymbol{y} - \hat{\boldsymbol{y}}\|_2^2 \ , \tag{3.2}$$

where $\hat{\boldsymbol{y}} = \boldsymbol{X}\boldsymbol{\beta}$. The optimal, i.e the ordinary least squares (OLS) solution, is

$$\boldsymbol{\beta}_{\text{OLS}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y} \ . \tag{3.3}$$

However, the OLS solution is not always satisfactory. For instance, the matrix $\boldsymbol{X}^T\boldsymbol{X}$ is not invertible when $d > N$, and $\boldsymbol{\beta}_{\text{OLS}}$ cannot even be evaluated.

The ridge regression is a well-known strategy to improve the OLS estimates (Hoerl and Kennard, 1970). The regression coefficients are estimated by minimizing the regularized MSE

$$\underset{\boldsymbol{\beta}}{\text{minimize}} \quad \frac{1}{N}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 + \lambda \sum_{i=1}^{d} \beta_i^2 \tag{3.4}$$

where $\lambda \geq 0$ is a tuning parameter. The optimal solution is

$$\boldsymbol{\beta}_{\mathrm{RR}} = (\boldsymbol{X}^T\boldsymbol{X} + \lambda\boldsymbol{I})^{-1}\boldsymbol{X}^T\boldsymbol{y}. \tag{3.5}$$

Due to the regularization term, the absolute values of the estimates $\boldsymbol{\beta}_{\mathrm{RR}}$ are shrunk toward zero in comparison to the OLS estimates. The amount of shrinkage is increased by increasing the value of λ. The shrinkage introduces bias to the estimates, but it decreases sensitivity of the estimates to the particular set of data being used. In addition, it may improve the prediction performance of the model.

3.2 Stepwise subset selection methods

The main drawback of the ridge regression estimates $\boldsymbol{\beta}_{\mathrm{RR}}$ in Equation (3.5) is that all the coefficients are typically nonzero regardless of the value of the tuning parameter λ (Fan and Li, 2001). Hence, all the variables are present in the model, although the importance of some inputs would be negligible. For interpretational reasons it would be beneficial to exclude these useless inputs from the model. Stepwise subset selection methods (Miller, 1990), such as forward selection and backward elimination, are traditional strategies to obtain candidate subsets of the input variables. The classical variants of these methods apply the OLS estimates that are evaluated using subsets of input variables.

3.2.1 Forward and backward selection

In the best subset regression, one finds the subset of $k \in \{0, \ldots, d\}$ inputs that minimizes the MSE. The subsets including k_1 and k_2 ($k_2 > k_1$) inputs need not be nested, that is, all the inputs in the smaller subset are not necessarily included in the larger one. The best subset selection is computationally infeasible for a large number of input variables due to the exponential complexity.

A forward stepwise selection strategy produces a sequence of nested subsets of inputs. It starts from an empty set of inputs and adds an input that most decreases the MSE in each subsequent step. The added input also has the largest absolute correlation with the current residuals of the model. Backward elimination operates in the opposite direction by deleting inputs one at the time starting from the full model. Obviously, the deleted input increases the MSE by the least amount. Both forward selection and backward elimination produce monotonic curves for the training error, which cannot be used in the selection of the optimal subset of inputs. Hastie et al. (2001) suggest to use the F-statistic to evaluate improvement in the fit between successive additions or deletions. However, it controls model selection only locally and it does not necessarily find the best subset from the candidates. A more secure approach is to select the final model based on an estimate of the generalization error.

If the OLS estimates are used, backward elimination can be applied only in the case of $N > d$, whereas the forward selection strategy can proceed as long as the number of selected inputs is less than observations. In both algorithms, decisions that are made in the early stages of the algorithm cannot be changed anymore. Somol et al. (1999) present an adaptive floating search method, in which the number of forward and backward steps are determined dynamically in the course of the algorithm. Nevertheless, Reunanen (2003) shows that the computationally more expensive floating search does not necessarily outperform the traditional forward selection algorithm in accuracy of the predictions.

3.2.2 Sequential input selection algorithm

A simple and efficient backward elimination method, called sequential input selection algorithm (SISAL), is proposed in Publication 4. In SISAL, the number of candidate subsets of input variables equals to the number of available inputs. It is clearly less than, for instance, in the traditional forward selection and backward elimination algorithms.

In Publication 4, SISAL is presented and applied to time series data. However, SISAL is not particularly designed to time series analysis and it can be used in any other regression setting as well. SISAL starts from the full OLS solution. A resampling technique, such as the bootstrap or CV, is applied to produce B replications of the regression coefficients $\beta_i, i = 1, \ldots, d$, which formulate the sampling distributions of the coefficients. The significance of each input variable is measured using the ratio

$$r_i = \frac{|m_{\beta_i}|}{\Delta_{\beta_i}} \quad , \tag{3.6}$$

where m_{β_i} is the median of the B replications. The width of the distribution is defined by the difference $\Delta_{\beta_i} = \beta_i^{\text{high}} - \beta_i^{\text{low}}$, where β_i^{high} and β_i^{low} are the $(1-\gamma)B^{\text{th}}$ and γB^{th} values in the ordered list of the B replications, respectively. The parameter $\gamma \in (0, 0.5)$ defines the central interval of the replications. The location and the width of the distribution are estimated using the median m_{β_i} and the difference Δ_{β_i}, since they are reasonable estimates for both asymmetric and symmetric distributions. In addition, they are insensitive to outliers. The input x_i corresponding to the smallest ratio r_i is discarded from the model. The resampling process is repeated using the remaining inputs as long as there are variables left to prune.

The proposed ratio r_i in Equation (3.6) is closely related to the traditional Z-score test statistic to test the hypothesis that a particular coefficient $\beta_i = 0$ (Hastie et al., 2001). Obviously, any quantity $q_i(|m_{\beta_i}|, \Delta_{\beta_i})$ that increases with the increasing absolute value of the median $|m_{\beta_i}|$ and decreases with the increasing width Δ_{β_i} could be used as a relevance measure. Thus, it is possible to adjust an effect of the values $|m_{\beta_i}|$ and Δ_{β_i} to the pruning of inputs, for instance, using a priori knowledge of the problem.

3.3 Input selection and shrinkage methods

A prediction method is called stable if the estimated regression equation does not change drastically with small changes in the data. Breiman (1996) shows that the subset selection strategies are unstable and the ridge regression is stable. Although the subset selection can be stabilized by averaging over several models, it may result in the usage of all the input variables. Simultaneous shrinkage and input selection methods pursue to combine the advantages of both subset selection and ridge regression, that is, sparsity and stability.

3.3.1 Nonnegative garrote

Breiman (1995) introduces nonnegative (NN) garrote to improve the subset selection and the ridge regression. The following optimization problem is considered

$$\begin{aligned} &\underset{\boldsymbol{c}}{\text{minimize}} && \sum_{n=1}^{N}(y_n - \sum_{i=1}^{d} c_i \beta_i^{\text{OLS}} x_{ni})^2 \\ &\text{such that} && \sum_{i=1}^{d} c_i \le t, \text{ and } c_i \ge 0, \ i = 1, \ldots, d \ . \end{aligned} \tag{3.7}$$

The NN garrote estimates $\beta_i^{\text{NNG}} = c_i \beta_i^{\text{OLS}}$ are scaled versions of the OLS estimates. When the hyperparameter t is decreased, more of the shrinkage factors c_i become zero and the remaining nonzero coefficients β_i^{NNG} are shrunk toward zero. For given t, the NN garrote solution is found by quadratic programming techniques. A drawback of the NN garrote is the explicit dependency of the OLS estimates. Instead of the OLS solution, Yuan and Lin (2007) suggest to use the ridge regression estimates as an initial solution for the NN garrote. Moreover, they show that the solution path of the NN garrote is piecewise linear and propose an efficient algorithm to build the whole path.

3.3.2 LASSO

Least absolute shrinkage and selection operator (LASSO) proposed by Tibshirani (1996) is closely related to the NN garrote. LASSO also shrinks some coefficients and sets others exactly to zero, but it does not rely on the OLS solution. The LASSO problem is defined by

$$\begin{aligned} &\underset{\boldsymbol{\beta}}{\text{minimize}} && \frac{1}{N}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 \\ &\text{such that} && \sum_{i=1}^{d} |\beta_i| \le t \ , \end{aligned} \tag{3.8}$$

which is a quadratic programming problem with a inequality constraint. The tuning parameter $t \ge 0$ controls the amount of shrinkage that is applied to the estimates. If $t \ge t_0 = \sum |\beta_i^{\text{OLS}}|$, the LASSO estimates coincide with the OLS solution. Otherwise, the estimates are shrunk toward zero, and with a small enough

t some of the coefficients are set exactly to zero. Thus, the LASSO performs a kind of continuous subset selection. The value of tuning parameter t should be chosen such that an estimate of the generalization error is minimized.

Tibshirani (1996) notes that the LASSO constraint $\sum |\beta_i| \leq t$ is equivalent to the addition of a regularization term $\lambda \sum |\beta_i|$ to the MSE cost function, i.e. there is a one-to-one correspondence between the parameters t and λ. However, the nondifferentiability of the constraints complicates the optimization in both cases. Schmidt et al. (2007) review and compare several optimization strategies to solve the regularized formulation for given λ. Osborne et al. (2000) show that the optimal solution trajectory of the LASSO problem is a piecewise linear as a function of t and propose an efficient algorithm to follow the trajectory.

Zou and Hastie (2005) introduce another simultaneous input selection and shrinkage method called elastic net. The regression coefficients are evaluated by minimizing MSE such that a weighted sum of the ridge $\sum \beta_i^2$ and LASSO $\sum |\beta_i|$ constraints is equal or less than a predefined threshold. The elastic net provides two extensions for the LASSO problem. First, elastic net is able to select an unknown group of highly correlated inputs, whereas LASSO tends to select only one variable from the group. Second, LASSO can select at most N variables in the $d > N$ case, but elastic net is able to include more than N variables to the model due to the grouping effect.

Bakin (1999) and Yuan and Lin (2006) consider the selection of known groups of non-overlapping input variables with a group LASSO problem. The d inputs are divided into the J groups, and MSE is minimized subject to a sparsity constraint as follows

$$\begin{aligned} \underset{\boldsymbol{\beta}}{\text{minimize}} \quad & \frac{1}{N}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 \\ \text{such that} \quad & \sum_{j=1}^{J} \|\overline{\boldsymbol{\beta}}_j\|_2 \leq t \ , \end{aligned} \tag{3.9}$$

where $\overline{\boldsymbol{\beta}}_j$ is a vector of regression coefficients of the jth group. With a small enough $t \geq 0$, some of the norms $\|\overline{\boldsymbol{\beta}}_j\|_2$ are zero and the corresponding groups of inputs are excluded from the model.

3.3.3 Least angle regression

Least angle regression (LARS) by Efron et al. (2004) is a less greedy version of the traditional forward selection algorithm. The LARS algorithm starts from the empty model, just like the traditional forward selection does. First, an input x_{i_1} that correlates most with the response is added to the model. The largest step possible in the direction of x_{i_1} is taken, until another input x_{i_2} is as correlated with the current residuals as x_{i_1} is. Forward selection would continue along x_{i_1}, but LARS proceeds in a direction equiangular between x_{i_1} and x_{i_2}, until the third input x_{i_3} is as correlated with the current residuals as the already added inputs are. LARS proceeds equiangularly between all the three added inputs until the fourth input can be added to the model. The algorithm continues similarly until all the inputs are selected. The model coincides with the OLS solution in the end.

Turlach (2004) gives an alternative description of the LARS algorithm. In each iteration, the correlations between all the inputs and the current residuals are evaluated. The input variables having the maximum absolute correlation are chosen and the OLS solution $\bar{y}_{k+1}$ is calculated using these inputs. The current response $\hat{y}_k$ is moved toward $\bar{y}_{k+1}$ until a new input will enter the model, i.e. its correlation with the residuals will coincide with the maximum value.

The estimates of the LARS algorithm are similar to the LASSO estimates. The sign of a nonzero LASSO estimate equals to the sign of the correlation between a corresponding input and the residuals of the model, but the LARS algorithm does not have this property. However, Efron et al. (2004) present also a modified version of the LARS algorithm that satisfy the previous property. Contrary to the original LARS algorithm, the modified version is also able to remove the already added inputs from the model. Furthermore, the modified LARS algorithm produces all the LASSO solutions if the inputs are added or removed one at a time.

3.3.4 Comparison with simulated data

The prediction performance and the correctness of input selection of SISAL, forward selection, and LARS are illustrated using simulated data sets that are generated according to the model in Equation (3.1). The number of samples and inputs are $N = 100$ and $d = 50$, respectively. Following Breiman and Friedman (1997), the rows $\underline{\boldsymbol{x}}_n$, $n = 1, \dots, N$, of the matrix $\boldsymbol{X}$ are sampled from a d-dimensional normal distribution

$$\underline{\boldsymbol{x}}_n \sim \mathrm{N}(\mathbf{0}, \boldsymbol{\Sigma}_x), \quad \text{where} \quad [\boldsymbol{\Sigma}_x]_{ij} = \sigma_x^{|i-j|} \ . \tag{3.10}$$

The parameter σ_x controls the amount of correlation between the input variables. The additive independently and normally distributed noise $\varepsilon_n \sim \mathrm{N}(0, 0.65^2)$ is used. A sparse structure is obtained by selecting randomly ten inputs. The regression coefficients of the selected ten inputs are drawn independently from a normal distribution with zero mean and unit variance, and the rest are set to zero. The prediction performance is measured by the mean squared estimation error

$$\mathrm{MSE} = (\boldsymbol{\beta}^* - \boldsymbol{\beta})^T \boldsymbol{\Sigma}_x (\boldsymbol{\beta}^* - \boldsymbol{\beta}) \ , \tag{3.11}$$

where $\boldsymbol{\beta}^*$ is the actual coefficient vector and $\boldsymbol{\beta}$ is the estimate of it (Breiman and Friedman, 1997).

Three values for the parameter σ_x are considered, i.e. $\sigma_x = 0$, $\sigma_x = 0.5$, and $\sigma_x = 0.9$. The inputs are uncorrelated with $\sigma_x = 0$, whereas in the case of $\sigma_x = 0.9$, the data contain nearly uncorrelated and highly correlated inputs. For each value of σ_x, the generation of data is repeated 500 times. In SISAL, the number of bootstrap replications is $B = 1000$.

On the left column of Figure 3.1, the average mean squared estimation errors are shown. The average number of correctly selected inputs are presented on the right column. The prediction accuracies of all the three methods are comparable with each other. The minimum error SISAL and forward selection models include always roughly ten inputs, but the number of correctly selected inputs decreases slightly

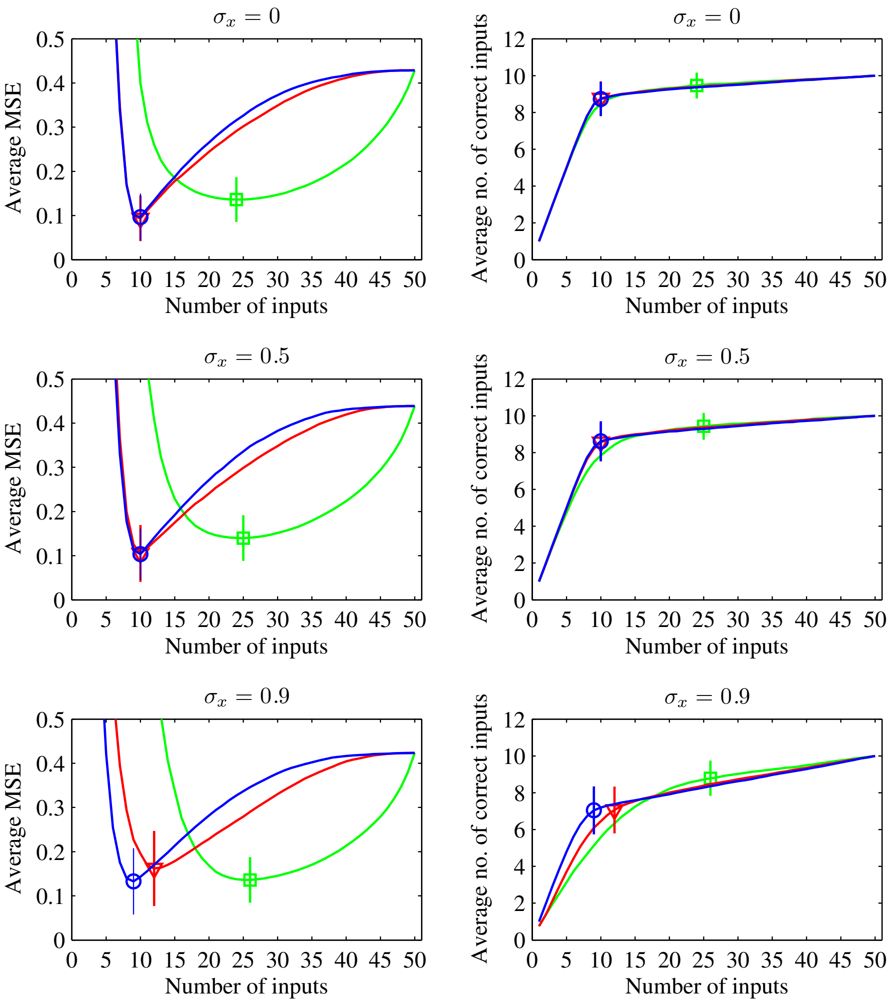

Figure 3.1: Average results of SISAL (*blue line*), forward selection (*red line*), and LARS (*green line*) for 500 replicates of simulated data with $\sigma_x = 0$ (*top row*), $\sigma_x = 0.5$ (*middle row*), $\sigma_x = 0.9$ (*bottom row*). The markers indicate the minimum prediction error models and the vertical lines represent the interval $[\mu - s, \mu + s]$, where μ is the mean and s is the standard deviation.

when the value of σ_x increase. The subsets of inputs obtained by LARS include more correct inputs, but the total number of selected inputs is approximately 25 with all the values of σ_x. The standard deviations of the minimum error models increase as the value of σ_x increase, especially in the cases of SISAL and forward selection.

3.4 Multiresponse linear regression

In a multiresponse linear regression problem, the objective is to estimate several response variables using a common set of inputs. Let us assume that there are N observations from q response variables $\boldsymbol{Y} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_q]$ and d input variables $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_d]$ for the model construction. A straightforward strategy is to construct q separate single response models. Even though all the separate models would be parsimonious in terms of the inputs, all the inputs may still be used. Alternatively, all the responses can be estimated simultaneously by a single model. Breiman and Friedman (1997) show by extensive simulations that simultaneous modeling may provide more accurate predictions than separate single response models, especially when the responses are correlated. In addition, sometimes the interest might be in finding a subset of inputs that can explain all the responses well (Barrett and Gray, 1994). Simultaneous estimation may also be computationally more efficient than separate modeling (Sparks et al., 1985).

The multiresponse linear regression model is defined as

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{W} + \boldsymbol{E} \ , \tag{3.12}$$

where $\boldsymbol{W}$ is a $d \times q$ matrix of regression coefficients and $\boldsymbol{E}$ is a noise matrix of size $N \times q$. The elements of the matrix $\boldsymbol{E}$ are assumed to be independently and normally distributed with zero mean and a common finite variance. It is further assumed that the columns of the matrices $\boldsymbol{X}$ and $\boldsymbol{Y}$ have zero mean, thus a bias term is not needed. Also, the scales of the columns are assumed to be similar. The OLS estimates solve the problem

$$\underset{\boldsymbol{W}}{\text{minimize}} \quad \|\boldsymbol{Y} - \hat{\boldsymbol{Y}}\|_F^2 = \|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{W}\|_F^2, \tag{3.13}$$

where $\|\cdot\|_F^2$ denotes the Frobenius norm, that is, the sum of squares of all the elements of the matrix. The OLS estimates are

$$\boldsymbol{W}_{\text{OLS}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{Y} \ . \tag{3.14}$$

Each column $\boldsymbol{w}_j, j = 1, \ldots, q$ of $\boldsymbol{W}_{\text{OLS}}$ coincides with the solution of the single response regression of $\boldsymbol{y}_j$ on $\boldsymbol{X}$.

As in the case of a single response variable, the prediction accuracy of the OLS estimates in Equation (3.14) can be improved by shrinkage methods. Brown and Zidek (1980) propose the multivariate version of the ridge regression. The Curds and Whey methods by Breiman and Friedman (1997) use canonical coordinates to shrink the OLS solution. In the case of high correlations among both the responses and the inputs, latent variable methods provide competitive models in terms of the prediction accuracy (Burnham et al., 1996; Abraham and Merola, 2005). Latent variables are constructed by linear combinations of the original variables by maximizing an appropriate objective function. The prediction of the response variables is further based on a small number of latent variables $d' < d$.

The inclusion of all the available original input variables into the model is the main drawback of shrinkage and latent variable models. In Equation (3.12), it is required that the row $\underline{\boldsymbol{w}}_i$ of $\boldsymbol{W}$ contains only zero elements such that the corresponding input $\boldsymbol{x}_i$ is excluded from the model. Sparks et al. (1985); Barrett and Gray (1994) introduce stepwise methods for input variable selection using criteria, that are generalized from the single response case. The criteria are based on the sum of squared errors and used to rank subsets of inputs, whereas MacKay (1977) discusses simultaneous statistical test procedures to select proper subsets of inputs.

3.4.1 Simultaneous variable selection methods

Simultaneous variable selection and shrinkage methods extending the LASSO problem in Equation (3.8) to the multiresponse case are introduced. The methods are formulated such that the minimization of the error in Equation (3.13) is constrained or penalized by the 1-norm of importance factors $\|\underline{\boldsymbol{w}}_i\|_\alpha$ of the input variables $\boldsymbol{x}_i$, $i = 1, \ldots, d$. The formulation results in a single convex optimization problem and the technique is called simultaneous variable selection (L_α-SVS).

Turlach et al. (2005) and Tropp (2006) consider the case $\alpha = \infty$, that is, the L_∞-SVS problem

$$\begin{aligned} \underset{\boldsymbol{W}}{\text{minimize}} \quad & \frac{1}{2}\|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{W}\|_F^2 \\ \text{such that} \quad & \sum_{i=1}^{d} \|\underline{\boldsymbol{w}}_i\|_\infty \leq t \ . \end{aligned} \tag{3.15}$$

The parameter $t \geq 0$ controls the amount of shrinkage and the number of selected input variables. When the value of t is decreased, more of the factors $\|\underline{\boldsymbol{w}}_i\|_\infty$ are set to zero, i.e the corresponding inputs are rejected from the model and the rest of the factors are shrunk toward zero. If $N > d$ and the input variables $\boldsymbol{x}_i$ are orthonormal, an efficient algorithm to follow the solutions of the L_∞-SVS problem as a function of t exists (Turlach et al., 2005). For the general case, Turlach et al. (2005) propose an efficient interior point algorithm. However, Turlach et al. (2005) suggest to use the problem in Equation (3.15) just as an explanatory tool to identify a subset of input variables, and subsequently estimate an unconstrained model using the selected inputs.

The L_2-SVS problem

$$\begin{aligned} \underset{\boldsymbol{W}}{\text{minimize}} \quad & \frac{1}{2}\|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{W}\|_F^2 \\ \text{such that} \quad & \sum_{i=1}^{d} \|\underline{\boldsymbol{w}}_i\|_2 \leq t \end{aligned} \tag{3.16}$$

is analyzed by Cotter et al. (2005) and Malioutov et al. (2005) and in Publication 3. Cotter et al. (2005) propose an M-FOCUSS algorithm to solve a penalized version of the L_2-SVS problem. In the M-FOCUSS algorithm, a sequence of weighted least squares problems is solved. Malioutov et al. (2005) also solve the penalized formulation. In contrast to the iterative M-FOCUSS algorithm, Malioutov et al. (2005) introduce an interior point algorithm in the second order cone programming framework.

A detailed analysis of the L_2-SVS problem is presented in Publication 3. The optimality conditions for the solution are derived. Furthermore, it is shown that the solution is unique if the inputs $\boldsymbol{x}_i$ corresponding to the nonzero factors $\underline{\boldsymbol{w}}_i$

are linearly independent. In Publication 3, a barrier method to solve the L_2-SVS problem for a fixed value of t is proposed. The implementation takes into consideration the structure of the problem reducing computational complexity of matrix inversions. In addition, an efficient predictor-corrector algorithm to evaluate the solution path for a given sequence of points $t_1 < \cdots < t_K$ is proposed. For each value of t_k, calculations are restricted to the active set of inputs, that is, only the rows $\underline{\boldsymbol{w}}_i$ of the matrix $\boldsymbol{W}$ that are likely to be nonzero are updated. In the case of a dense sequence, the current solution $\boldsymbol{W}_{t_k}$ is an appropriate initialization for the next value t_{k+1}, but it can also be improved, if necessary, by a simple scaling procedure.

Tropp (2006) reports that no empirical or theoretical evidence exists that would substantially support one problem formulation over another for simultaneous variable selection. In publication 3, L_2-SVS and L_∞-SVS are compared to each other using simulated and real data sets. In the case of real data, L_2- and L_∞-SVS are comparable in terms of the prediction accuracy and the number of selected input variables. As Turlach et al. (2005) suggest, the subset OLS solutions are also evaluated. The subset OLS models result in the similar minimum prediction errors as the estimates of the L_2- and L_∞-SVS problems. However, the subset OLS models include smaller numbers of inputs than the L_2- and L_∞-SVS models. The amount of correlation between the inputs is varied in simulated data sets and L_2-SVS is found to perform better than L_∞-SVS in terms of both the prediction accuracy and the correctness of input selection.

Similä (2007b) introduces the following penalized least squares problem

$$\underset{\boldsymbol{W}}{\text{minimize}} \quad \frac{1}{2}\|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{W}\|_F^2 + \lambda \sum_{i=1}^{d} p(\|\underline{\boldsymbol{w}}_j\|_2) \tag{3.17}$$

for simultaneous variable selection. It is an extension of the work by Fan and Li (2001) for a one-dimensional response space ($q = 1$) to the case of multiple responses ($q > 1$). The penalty function $p(\|\underline{\boldsymbol{w}}_j\|_2)$ can be any increasing, differentiable, and concave function. Particularly, strictly concave penalty functions are needed to avoid the bias in the case of large true unknown factors $\|\underline{\boldsymbol{w}}_j\|_2$. Similä (2007b) proposes a majorize-minimize algorithm to solve the problem and also suggests an active set strategy for a sequence of descending values of the regularization parameter λ. In experiments, he uses a penalty function of the form

$$p(\|\underline{\boldsymbol{w}}_j\|_2) = c \log\left(1 + \frac{\|\underline{\boldsymbol{w}}_j\|_2}{c}\right) , \tag{3.18}$$

where the degree of concavity is increased by decreasing the parameter $c > 0$. It is reported that models are equally accurate regardless of the value of c. However, with smaller values of c the total number of selected input variables decreases, while the number of relevant ones remains approximately unchanged.

3.4.2 The multiresponse sparse regression algorithm

In Publications 1 and 2, the multiresponse sparse regression (MRSR) algorithm is proposed. It is an extension of the LARS algorithm to the case of multiple

Algorithm 1 L_α-MRSR

```
1: Set $k = 0$, $\hat{\boldsymbol{Y}}_k = \boldsymbol{0}$, and $\hat{\boldsymbol{W}}_k = \boldsymbol{0}$, define $k_{\max} \le \min\{d, N-1\}$
2: Calculate
   $c_{k,i} = \|(\boldsymbol{Y} - \hat{\boldsymbol{Y}}_k)^T \boldsymbol{x}_i\|_\alpha$,  and  $c_k^{\max} = \max_i\{c_{k,i}\}$
3: Define the active set of inputs $\mathcal{A}_{k+1} = \{i : c_{k,i} = c_k^{\max}\}$
4: Calculate the OLS solution using $\boldsymbol{X}_{k+1} = [\cdots \boldsymbol{x}_i \cdots]_{i \in \mathcal{A}_{k+1}}$:
   $\boldsymbol{W}_{k+1}^{\mathrm{OLS}} = (\boldsymbol{X}_{k+1}^T \boldsymbol{X}_{k+1})^{-1} \boldsymbol{X}_{k+1}^T \boldsymbol{Y}$,  and  $\boldsymbol{Y}_{k+1}^{\mathrm{OLS}} = \boldsymbol{X}_{k+1} \boldsymbol{W}_{k+1}^{\mathrm{OLS}}$
5: Update the MRSR estimates
   $\hat{\boldsymbol{W}}_{k+1} = \hat{\boldsymbol{W}}_k + \gamma_k(\bar{\boldsymbol{W}}_{k+1}^{\mathrm{OLS}} - \hat{\boldsymbol{W}}_k)$,  and  $\hat{\boldsymbol{Y}}_{k+1} = \hat{\boldsymbol{Y}}_k + \gamma_k(\boldsymbol{Y}_{k+1}^{\mathrm{OLS}} - \hat{\boldsymbol{Y}}_k)$,
   If $|\mathcal{A}_{k+1}| < d$
      the step length $\gamma_k$ is defined such that a new input $\boldsymbol{x}_i, i \notin \mathcal{A}_{k+1}$ is added
      to the active set in the next iteration, see details in the text.
   Else
      $\gamma_k = 1$
6: Set $k = k + 1$, and go to the 2nd step if $k < k_{\max}$
```

responses. The forward selection strategy is adopted and the selected inputs provide accurate predictions averaged over all the response variables. MRSR takes careful steps toward the OLS solution in the spirit of the LARS algorithm. Thus, it combines input selection and shrinkage of the regression coefficients.

The MRSR algorithm is presented in detail in Algorithm 1. It is assumed that the variables $\boldsymbol{X}$ and $\boldsymbol{Y}$, the inputs of the algorithm, have zero mean and comparable scales. The MRSR estimates, the outputs of the algorithm, are initialized to zero $\hat{\boldsymbol{W}}_k = \boldsymbol{0}$ and $\hat{\boldsymbol{Y}}_k = \boldsymbol{0}$ and the maximum number of iterations $k_{\max}$ is defined. The second step is to evaluate the correlation between each input variable $\boldsymbol{x}_i$ and the residuals $(\boldsymbol{Y} - \hat{\boldsymbol{Y}}_k)^T$, where $\hat{\boldsymbol{Y}}_k$ is the current MRSR estimate of the responses. The inputs having the maximum correlation $c_k^{\max}$ define the active set $\mathcal{A}_{k+1}$ of inputs (Step 3). In the next step, the OLS estimates $\boldsymbol{W}_{k+1}^{\mathrm{OLS}}$ and $\boldsymbol{Y}_{k+1}^{\mathrm{OLS}}$ are calculated using only the active inputs $\mathcal{A}_{k+1}$. In Step 5, the MRSR estimates $\hat{\boldsymbol{W}}_{k+1}$ and $\hat{\boldsymbol{Y}}_{k+1}$ are updated. The nonzero rows of the $d \times q$ sparse matrix $\bar{\boldsymbol{W}}_{k+1}^{\mathrm{OLS}}$ are indexed by $\mathcal{A}_{k+1}$ and they are equal to the corresponding rows of $\boldsymbol{W}_{k+1}^{\mathrm{OLS}}$. If the updating step length was always $\gamma_k = 1$, the MRSR algorithm would coincide with the traditional forward selection approach. Therefore, the choice $\gamma_k \in (0, 1)$ shrinks the values of coefficients of the active inputs $\mathcal{A}_{k+1}$ and keeps the coefficients of the nonactive inputs zero. The current estimate $\hat{\boldsymbol{Y}}_k$ is moved toward the OLS solution $\boldsymbol{Y}_{k+1}^{\mathrm{OLS}}$ until one of the nonactive inputs is as correlated with the residuals as the active inputs. In the next iteration, the correlations are

$$\begin{array}{ll} c_{k+1,i}(\gamma) = (1-\gamma)c_k^{\max} & \text{for } i \in \mathcal{A}_{k+1} \text{ and} \\ c_{k+1,i}(\gamma) = \|\boldsymbol{u}_{k,i} - \gamma \boldsymbol{v}_{k,i}\|_\alpha & \text{for } i \notin \mathcal{A}_{k+1}\ , \end{array} \tag{3.19}$$

where $\boldsymbol{u}_{k,i} = (\boldsymbol{Y} - \hat{\boldsymbol{Y}})^T \boldsymbol{x}_i$ and $\boldsymbol{v}_{k,i} = (\boldsymbol{Y}_{k+1}^{\mathrm{OLS}} - \hat{\boldsymbol{Y}}_k)^T \boldsymbol{x}_i$. An input $\boldsymbol{x}_i$, $i \notin \mathcal{A}_{k+1}$, is added to the model when the statements in Equation (3.19) are equal. The correct step length γ_k is the smallest of the candidate step lengths. Steps 2-5 are repeated until the maximum number of iterations $k_{\max}$ is performed. If $k_{\max} = d < N - 1$, the step length $\gamma_k = 1$ is applied in the last iteration and the MRSR solution

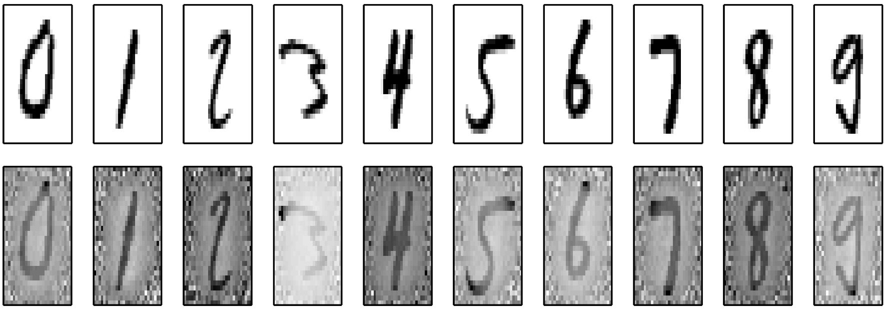

Figure 3.2: Examples of original unnoisy images of handwritten digits (*top row*) and the corresponding normalized noisy images (*bottom row*).

coincides with the OLS solution.

The MRSR algorithm with $\alpha = 1$ is proposed in Publication 1. However, the proposed strategy to evaluate the correct step length γ_k is inefficient in the case of a large number of response variables. In Publication 2, the MRSR algorithm with the choices $\alpha \in \{1, 2, \infty\}$ is analyzed. For each choice of α, a step length evaluation procedure that is also efficient for a high-dimensional response space, is presented. It is also shown in Publication 2, that there always exists such a step length $\gamma \in (0, 1]$ for each $i \notin \mathcal{A}_{k+1}$ that the common correlation curve $(1-\gamma)c_k^{\max}$ and the correlation curve $\|\boldsymbol{u}_{k,i} - \gamma \boldsymbol{v}_{k,i}\|_\alpha$ of a nonactive input $\boldsymbol{x}_i$ intersect.

Yuan and Lin (2006) introduce the group LARS algorithm to approximate the solution path of the group LASSO problem in Equation (3.9). The connection between the group LARS and L_2-MRSR algorithms is discussed in Publication 2, that is, with rearrangement of $\boldsymbol{Y}$, $\boldsymbol{X}$, and $\boldsymbol{W}$ the outcomes of the algorithms are equal. Similä (2007a) establishes the connection between the L_2-MRSR algorithm and the L_2-SVS problem showing that L_2-MRSR follows the solution path of the L_2-SVS problem under the assumptions $\boldsymbol{X}^T\boldsymbol{X} = \boldsymbol{I}$. For general $\boldsymbol{X}$ and $q > 1$, the solution path of the L_2-SVS problem is nonlinear as a function of t, and the L_2-MRSR algorithm only approximates the path.

Illustration of the MRSR algorithm

The image data reconstruction experiment from Publication 2 is revisited. A sample of handwritten digits $0, \ldots, 9$, each a 28×28 gray scale image, is analyzed. Each image is represented as a vector $\boldsymbol{x}_n = [x_{n,1}, \ldots, x_{n,784}]$ containing the gray scale values of pixels. The data set includes 100 observations from each digit (total $N = 1000$ images). An equal amount of noise as in Publication 2 is added and each of the $d = 784$ variables is scaled to have zero mean and unit variance. The data are illustrated in Figure 3.2.

The original images are reconstructed from the noisy ones using the MRSR algorithm such that $\boldsymbol{X}$ and $\boldsymbol{Y}$ contain the same set of data. Since $\hat{\boldsymbol{W}}_k$ is row sparse, only some of the pixels are used in the reconstruction $\hat{\boldsymbol{Y}}_k$. The selected input

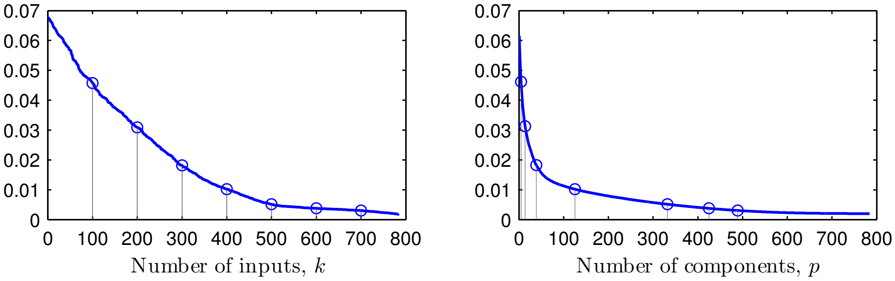

Figure 3.3: Mean squared reconstruction errors as a function of the number of inputs in the L_1-MRSR model (*left*) and the number of principal components (*right*).

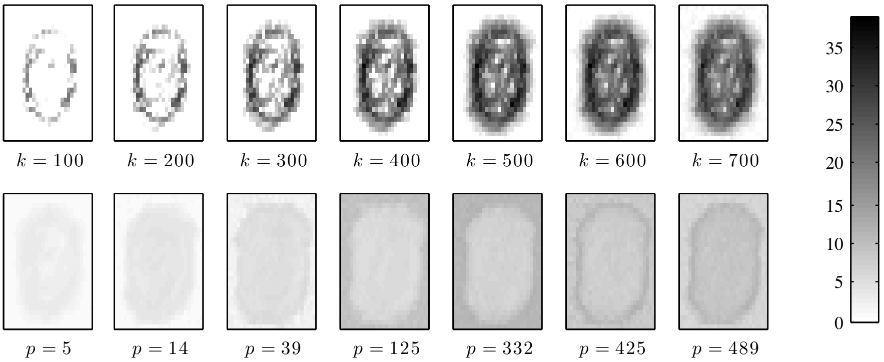

Figure 3.4: Importance of the original inputs measured using the factors $\|\underline{\boldsymbol{w}}_i\|_1$ (*top row*) and $\|\underline{\tilde{\boldsymbol{q}}}_i\|_1$ (*bottom row*) in the L_1-MRSR models and in the PCA models, respectively.

variables can be considered as principal variables of the data set. The principal variables preserve information of the data as accurately as possible according to some criterion (Cumming and Wooff, 2007). Here, goodness of the selected variables is evaluated using MSE.

PCA is an effective technique for dimensionality reduction by a linear combination of input variables. For a given dimensionality p, the reconstruction error

$$\|\boldsymbol{X} - \boldsymbol{X}\tilde{\boldsymbol{Q}}\|_F^2 \ , \tag{3.20}$$

where $\tilde{\boldsymbol{Q}} = \boldsymbol{Q}\boldsymbol{Q}^T$, is minimized under the orthonormality constraint $\boldsymbol{Q}^T\boldsymbol{Q} = \boldsymbol{I}_p$ if the columns of $\boldsymbol{Q}$ are the loadings of the principal components of $\boldsymbol{X}$ corresponding to the p largest eigenvalues (Haykin, 1999; Hastie et al., 2001).

The reconstruction errors for the L_1-MRSR models (*left*) and the PCA models (*right*) are shown in Figure 3.3. Only the errors for L_1-MRSR are shown, since it has the smallest errors among the choices $\alpha \in \{1, 2, \infty\}$ in this task (see Publication 2). In Figure 3.3, the vertical lines and circles mark the points in which the errors of both methods are equal. In these points, the dimension of the PCA model is smaller than the dimension of the L_1-MRSR model, i.e $p < k$. The importance factors of the original inputs are presented in Figure 3.4. The L_1-MRSR algorithm selects reasonable inputs, since relevant information is in the middle of the images. In the PCA models, none of the factors $\|\underline{\tilde{\boldsymbol{q}}}_i\|$ is ever exactly zero. In addition, the inputs corresponding to the borders of the images are considered to be the most important ones with the values $p = 135$ and $p = 332$. Thus, these PCA models give a misleading impression of importance of the original input variables. Similä (2007a) uses the problem in Equation (3.17) to detect the principal variables, and finds that the inputs enter the model in the correct order when the value of λ is decreased.

Chapter 4

Variable selection in nonlinear regression

Despite all the nice features of the linear regression models, they are not appropriate in every problem. Simply, the assumption on linear dependency between the inputs and the output is overly restrictive in some real world applications. For example, the kinetic energy of an object increases with the square of the speed. Unfortunately, the underlying input-output relationship is typically unknown in data analysis tasks. In the nonlinear regression problem, the goal is to estimate an unknown function f in the model

$$y = f(\boldsymbol{x}) + \varepsilon \ , \tag{4.1}$$

where ε is random additive noise with zero mean and finite variance.

The estimation of the function f is based on the observed data points $\{\boldsymbol{x}_n, y_n\}_{n=1}^{N}$ and the accuracy of f is commonly measured by the mean squared error

$$\text{MSE} = \frac{1}{N}\sum_{n=1}^{N}(y_n - f(\boldsymbol{x}_n))^2 \ . \tag{4.2}$$

However, for an arbitrary function f, the minimization of the error leads to infinitely many solutions (Hastie et al., 2001). Any function f that interpolates exactly the training points is a solution producing zero error. On the other hand, these models would probably have large errors for novel test data. To obtain a smooth mapping, some restrictions on the function f must be introduced. However, it does not totally remove the problem of the multiple solutions, since different constraints lead to different models that may be equally accurate (Hastie et al., 2001).

Nonlinear function estimation techniques can be divided into two classes, namely nonadaptive and adaptive methods (Cherkassky and Mulier, 1998). Nonadaptive methods are generally computationally fast, since the models consist of unadjustable predefined basis functions. Adaptive methods are more flexible, since the basis functions depend nonlinearly on parameters that are optimized based on the data. Due to the nonlinearities, iterative optimization strategies have to be used which increases the computational complexity. Nevertheless, in the present thesis the objective is to build interpretable adaptive models. The level of interpretability is hard to judge, but explaining dependencies of highly complex models is obviously difficult. Thus, the complexity of the model should be limited as much as possible without sacrificing the approximation accuracy considerably.

4.1 Nonadaptive methods

Specht (1991) presents a general regression neural network (GRNN)

$$f(\boldsymbol{x}) = \frac{\sum_{n=1}^{N} y_n \exp(-\frac{\|\boldsymbol{x}_n - \boldsymbol{x}\|^2}{\sigma^2})}{\sum_{n=1}^{N} \exp(-\frac{\|\boldsymbol{x}_n - \boldsymbol{x}\|^2}{\sigma^2})} \quad . \tag{4.3}$$

The model output is essentially a weighted average of all the observations y_n, where the weights are defined by the basis functions in the input space. The smoothing parameter σ controls the width of the local neighborhood. For a given value of σ, the training of the GRNN model is very fast since only the basis functions need to be evaluated. The fast training makes forward selection or backward elimination of the inputs applicable. The estimate is also known as the Nadaraya-Watson weighted average (Hastie et al., 2001), and instead of the Gaussian functions other weighting schemes can be used as well (Parzen, 1962; Atkeson et al., 1997).

Alternatively, the function f can be modeled by a linear expansion

$$f(\boldsymbol{x}) = \sum_{m=1}^{M} w_m h_m(\boldsymbol{x}) + w_0 \quad , \tag{4.4}$$

where $h_m(\boldsymbol{x})$ is a function of the input $\boldsymbol{x}$. Cherkassky and Mulier (1998) introduce polynomials and splines and Antoniadis et al. (1994) suggest wavelets to be used as the basis functions $h_m(\boldsymbol{x})$. After the basis functions are evaluated, the function $f(\boldsymbol{x})$ depends linearly on the parameters w_m. The number of available basis functions is typically very large and the regularization techniques should be used in the estimation of the parameters w_m (Donoho and Johnstone, 1994; Girosi et al., 1995). The basis functions may also include tuning parameters that have to be selected using some model selection criteria.

The parameters w_m can be estimated by sparse linear regression techniques that are presented in Chapter 3. However, sparsity is not achieved in terms of the input variables if all the basis functions depend on all the inputs. In the multivariate adaptive regression splines (MARS) model by Friedman (1991), piecewise linear basis functions are used. The original basis functions and products of them are added into the model by alternating the forward selection and backward elimination procedures. The resulting MARS model may or may not be sparse in terms of the original input variables. Lin and Zhang (2006) propose component selection and smoothing operator (COSSO) for model selection and model fitting in the framework of smoothing spline analysis of variance. In the COSSO model, the sum of component norms of the spline model is penalized such that some of the components are discarded from the final model. The original input variable is not included into the model if all the components depending on the corresponding input are discarded.

4.2 Support vector machine

Boser et al. (1992) propose the support vector machine (SVM) method for the classification problem and Drucker et al. (1997) extend it to the regression problem. A detailed description of SVM for both cases is given, for instance, by Cristianini and Shawe-Taylor (2000). Basically, the prediction error and complexity of the model are minimized simultaneously in the construction of the SVM model.

The derivation of SVM starts from the model

$$f(\boldsymbol{x}) = \boldsymbol{z}^T \psi(\boldsymbol{x}) + \alpha_0 \ , \tag{4.5}$$

where $\psi(\cdot)$ is a mapping from the input space $\boldsymbol{x}$ to a high (possible even infinite) dimensional feature space. Instead of the classical MSE, the ϵ-intensive cost function

$$|y - f(\boldsymbol{x})|_\epsilon = \begin{cases} 0 & \text{if } |y - f(\boldsymbol{x})| \leq \epsilon \\ |y - f(\boldsymbol{x})| - \epsilon & \text{otherwise} \end{cases} \tag{4.6}$$

is utilized. The errors that are smaller than ϵ do not contribute to the cost function at all.

With a data set $\{\boldsymbol{x}_n, y_n\}_{n=1}^N$, the following constrained optimization problem can be formulated

$$\begin{array}{ll} \underset{\boldsymbol{z},\alpha_0,\boldsymbol{\xi},\hat{\boldsymbol{\xi}}}{\text{minimize}} & \frac{1}{2}\boldsymbol{z}^T\boldsymbol{z} + C\sum_{n=1}^{N}(\xi_n + \hat{\xi}_n) \\ \text{such that} & y_n - \boldsymbol{z}^T\psi(\boldsymbol{x}_n) - \alpha_0 \leq \epsilon + \xi_n, \quad n = 1,\ldots,N \\ & \boldsymbol{z}^T\psi(\boldsymbol{x}_n) + \alpha_0 - y_n \leq \epsilon + \hat{\xi}_n, \quad n = 1,\ldots,N \\ & \xi_n, \hat{\xi}_n \geq 0, \quad n = 1,\ldots,N \end{array} \tag{4.7}$$

where ξ_n and $\hat{\xi}_n$ are the slack variables and the regularization parameter $C > 0$ controls the trade-off between prediction error and model complexity. In practice, the solution is found by formulating the dual problem and the resulting quadratic programming (QP) problem is solved. The sequential minimal optimization (SMO) algorithm for the regression problem by Flake and Lawrence (2002) is an efficient method to find the solution without a QP solver. The resulting model is

$$f(\boldsymbol{x}) = \sum_{n=1}^{N} \alpha_n K(\boldsymbol{x}_n, \boldsymbol{x}) + \alpha_0 \ , \tag{4.8}$$

where Lagrange multipliers α_n^+ and α_n^- are substituted by $\alpha_n = \alpha_n^+ - \alpha_n^-$ and the kernel trick $K(\boldsymbol{x}_n, \boldsymbol{x}) = \psi(\boldsymbol{x}_n)^T\psi(\boldsymbol{x})$ is applied. Due to the kernel trick, explicit calculations are not needed in the feature space. Any function that satisfies Mercer's conditions can be used as a kernel (Cristianini and Shawe-Taylor, 2000). The Gaussian kernel $K(\boldsymbol{x}_n, \boldsymbol{x}) = \exp(-\|\boldsymbol{x}_n - \boldsymbol{x}\|^2/\sigma^2)$, that corresponds to an infinite dimensional feature space, is perhaps the most typical choice. Smola and Schölkopf (2004) introduce several other options.

There are two major advantages in the SVM model. First, the optimization problem is convex for given values of C, ϵ, and kernel parameters. Convexity ensures global optimality of the solution. Second, usage of the ϵ-intensive cost function enables sparseness of the solution with respect to the parameters α_n. The drawback is that appropriate values for C, ϵ, and kernel parameters should be determined using cross-validation or another model selection criterion, which increases computational burden.

In the kernel ridge regression (KRR) by Saunders et al. (1998) and in the least squares support vector machine (LS-SVM) by Suykens, Van Gestel, De Brabanter, De Moor and Vandewalle (2002), the following optimization problem is considered

$$\begin{array}{ll} \underset{\boldsymbol{z},\alpha_0,\boldsymbol{e}}{\text{minimize}} & \frac{1}{2}\boldsymbol{z}^T\boldsymbol{z} + C\sum_{n=1}^{N} e_n^2 \\ \text{such that} & y_n = \boldsymbol{z}^T\psi(\boldsymbol{x}_n) + \alpha_0 + e_n, \quad n = 1,\ldots,N \ . \end{array} \tag{4.9}$$

In practice, it is a ridge regression formulation in the feature space. As in the case of SVM, the kernel trick is used to avoid explicit calculations in the feature space. The solution for the problem in Equation (4.9) is again evaluated using the dual formulation. The resulting LS-SVM model can be written as the SVM model in Equation (4.8). Now, the Lagrange multipliers α_n are found by solving a system of linear equations which is computationally more efficient than solving the QP problem in the training of the SVM model. However, it is very unlikely that any of the coefficients α_n would be zero in the LS-SVM model. The LS-SVM model includes one tuning parameter less than the standard SVM model, which decreases further the computational load.

Cawley and Talbot (2002) present a reduced rank kernel ridge regression (RRKRR) method that generates a sparse kernel expansion. The method consists of two parts. First, such a subset of training samples $\{\boldsymbol{x}_j\}_{j\in\mathcal{S}} \subset \{\boldsymbol{x}_n\}_{n=1}^N$ is found that the mappings of all the training samples into the feature space $\psi(\boldsymbol{x}_n)$ are approximated by a linear combination of the mappings of the samples in $\mathcal{S}$. The search starts from the empty set and samples are added one at a time such that the reconstruction error is minimized in each step (Baudat and Anouar, 2001). Second, a ridge regression model is constructed in the reduced space $\boldsymbol{z} = \sum_{j\in\mathcal{S}} \beta_j\psi(\boldsymbol{x}_j)$. Espinoza et al. (2006) propose an alternative strategy to build a sparse model. They maximize an entropy criterion to select $M \ll N$ support vectors. Furthermore, the mappings $\psi(\boldsymbol{x})$ are approximated using the selected support vectors and the primal problem in Equation (4.9) is solved directly. Several other techniques to approximate the kernel matrix $\boldsymbol{K} = \{K(\boldsymbol{x}_i, \boldsymbol{x}_j)\}_{i,j=1,\ldots,N}$ with a sparse lower rank matrix are presented by Smola and Schölkopf (2000).

The output observations y_n can also be taken into account in the construction of a sparse LS-SVM model. Suykens, De Brabanter, Lukas and Vandewalle (2002) suggest to train the ordinary LS-SVM model first. The support vectors corresponding to the smallest absolute values of the parameters α_n are omitted and the model is retrained using the reduced training set. Learning algorithms by Nair et al. (2002) operate in the opposite direction. At each iteration, a support vector is added based on the current residuals of the model. In addition, it is monitored that all the added support vectors are useful in the later stages of the algorithm. Hoegaerts et al. (2004) compare several pruning methods for LS-SVM and conclude that pruning based on the values of parameters α_n results in adequate compromise between accuracy and computational cost.

4.2.1 Input variable selection for the SVM

All of the SVM and LS-SVM models presented in the previous section are fully dense in terms of the input variables. It can deteriorate generalization capability of the SVM model as shown, for instance, by Weston et al. (2001). Bi et al. (2003) select inputs by constructing several sparse linear SVM models. The inputs having on average nonzero weights are further used in the training of a nonlinear SVM model. François et al. (2007) suggest to take nonlinear dependencies between the inputs into account by introducing a mutual information (MI) criterion for the input selection. They combine the MI criterion with a stepwise search strategy to find a proper subset. Obviously, any nonlinear prediction model, e.g. SVM or LS-SVM, can be built using the obtained subset of inputs.

Guyon et al. (2002) present a recursive feature elimination (RFE) algorithm for the classification problem, but it can be extended to the regression problem as well. First, the model is trained using all the input variables. Second, the inputs are ranked by evaluating the change in the cost function when one input at a time is removed from the model. The input causing the smallest change is the least relevant and eliminated. The selection strategy produces a list of ranked inputs. Rakotomamonjy (2003, 2007) extends the RFE algorithm by modifying the ranking criterion. The derivatives of the cost function with respect to the inputs are used. The least informative input has the smallest average absolute derivatives. In addition to the standard cost function, Rakotomamonjy (2003, 2007) measures sensitivity of the inputs with respect to different generalization error bounds.

Several authors consider adaptive scaling of the input variables (Weston et al., 2001; Van Gestel et al., 2001; Chapelle et al., 2002; Grandvalet and Canu, 2003; Rakotomamonjy, 2007). This is done by applying a kernel of the form

$$K_{\boldsymbol{w}}(\boldsymbol{x}_n, \boldsymbol{x}) = K(\boldsymbol{W}\boldsymbol{x}_n, \boldsymbol{W}\boldsymbol{x}) \ , \tag{4.10}$$

where $\boldsymbol{W} = \operatorname{diag}(w_1, \dots, w_d)$ is a diagonal matrix of scaling factors. Each input variable x_i has its own factor or weight $w_i > 0$. For instance, the Gaussian kernel with the adaptive weights is

$$K_{\boldsymbol{w}}(\boldsymbol{x}_n, \boldsymbol{x}) = \exp\left(-\sum_{i=1}^{d} w_i (x_{ni} - x_i)^2\right) \ . \tag{4.11}$$

Now, the problem of selecting inputs is equivalent to finding those weights w_i that can be set to zero without impairing the generalization performance considerably.

An alternating optimization strategy (Csiszár and Tusnády, 1984) is typically applied to perform adaptive scaling. First, the standard model is trained keeping the weights w_i fixed. Second, the weights w_i are optimized keeping the model otherwise fixed. These two steps are repeated as long as the minimum of the cost function is reached or the maximum number of iterations is performed. In the second step, Weston et al. (2001); Chapelle et al. (2002); Rakotomamonjy (2007) minimize directly bounds on generalization error estimates and propose to delete the inputs that have the smallest weights. The approach by Grandvalet and Canu (2003) differs in the optimization of the weights w_i. They restrict the sum of the weights by a predefined constant t and minimize the standard cost function under that constraint. The purpose of the constraint is to encourage sparsity in terms of the inputs. The generalization error is estimated for the several choices of the shrinkage parameter t and other hyperparameters.

Van Gestel et al. (2001) introduce adaptive scaling of the inputs in the LS-SVM model in the probabilistic framework. The weights are inferred by maximizing the model evidence under the assumption of uniform prior on the weights. The input having the smallest weight is rejected and the process is repeated using the remaining inputs.

4.3 Feedforward neural networks

An artificial neural network (ANN) is a powerful information processing system (Bishop, 1995; Haykin, 1999). Originally, the development of ANNs was inspired by biological systems (Haykin, 1999). In the present thesis, ANNs are merely considered as highly adaptive nonlinear mathematical models and learning is the adaptation of model parameters based on the data. Lately, ANNs have been applied successfully in different data analysis problems, for instance, in analyses of financial (Roberts, 2008) and ecological (Tomenko et al., 2007) data, simulation of biochemical systems (Matsubara et al., 2006), classification of gene expression data (Pirooznia et al., 2008), solving linear integro-differential equations (Golbabai and Seifollahi, 2007), and voltage stability monitoring of power systems (Chakrabarti and Jeyasurya, 2007).

The black box characteristics is a disadvantage of ANNs in the sense that interpretation of dependencies between the input and output variables is difficult. In critical applications, such as medical decision support, a network has to be understood thoroughly before it can be deployed (Dayhoff and DeLeo, 2001). Benítez et al. (1997); Kolman and Margaliot (2005) present rule extraction from the trained network by expressing the network in terms of simple fuzzy rules. Nonetheless, the number of extracted rules might be prohibitive in complex problems. In this work, several input selection algorithms are proposed to improve interpretability of ANNs. The algorithms are presented in Sections 4.3.2 and 4.3.3.

A neural network consists of input, hidden, and output layers. A network with a single hidden layer is presented in the top panel of Figure 4.1. The network is fully connected, i.e. each node in any layer of the network is connected to all the nodes in the previous layer. In the feedforward network, feedback loops are not allowed. The bottom panel of the same figure shows a widely applied structure of the hidden node.

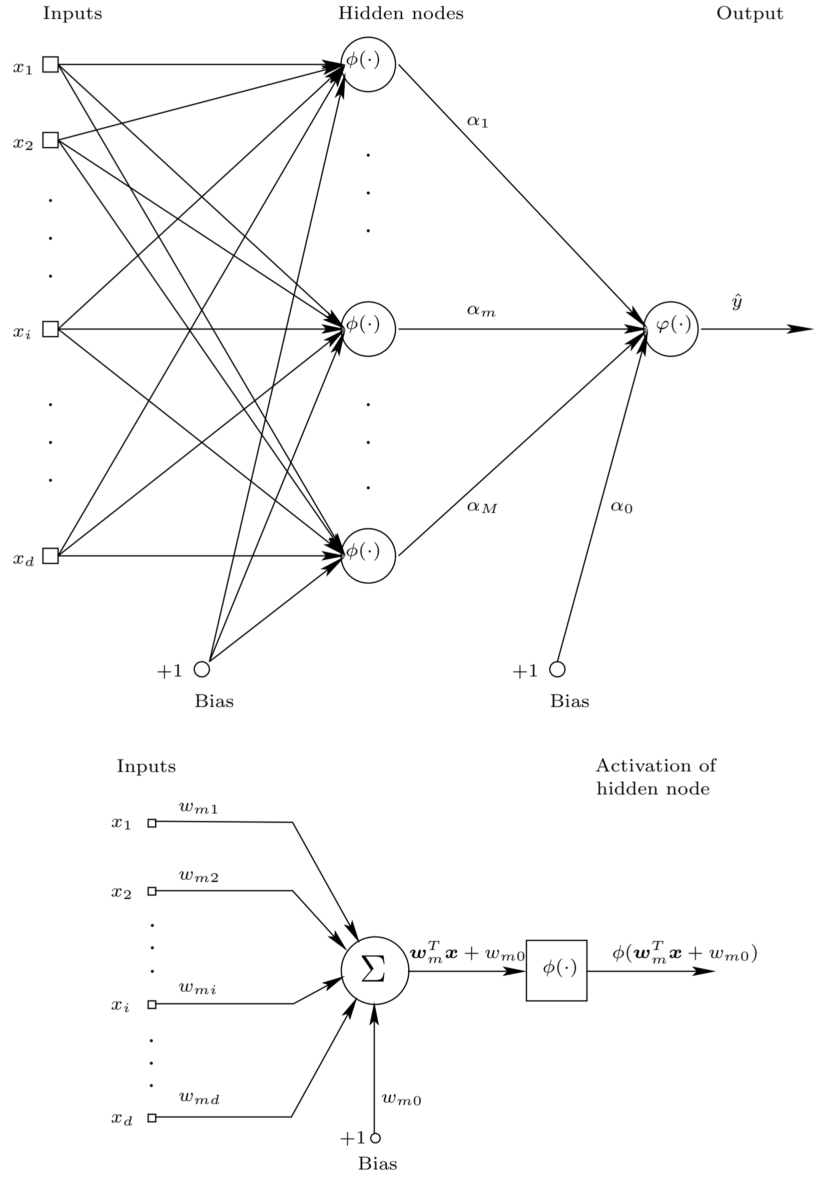

Figure 4.1: The architecture of feedforward neural network (*top*). Structure of a hidden node in the MLP network (*bottom*)

The input nodes correspond to the input variables x_i, $i = 1, \ldots, d$. The network output $\hat{y}$ is typically one-dimensional in the regression problems, but the generalization to the multidimensional case is straightforward (Hastie et al., 2001). The number of nodes in the hidden layer affect complexity of the network (Haykin, 1999). In the present thesis, networks with only one hidden layer are considered since they are universal approximators, i.e. capable to approximate any continuous smooth function with an arbitrary accuracy if the number of hidden nodes is large enough (Hornik et al., 1989, 1990; Park and Sandberg, 1991, 1993). However, a sufficient number depends highly on the problem and the generalization error estimation techniques are used to find an appropriate value in practice.

In Figure 4.1, the output of the network is

$$f(\boldsymbol{x}) = \varphi\left(\sum_{m=1}^{M} \alpha_m \phi(v_m) + \alpha_0\right) , \tag{4.12}$$

where M is the number of hidden nodes, $\phi(\cdot)$ and $\varphi(\cdot)$ are the activation functions of the hidden and output layers, respectively. In the case of multilayer perceptron (MLP) network (Rumelhart et al., 1986), the activation of each hidden node depends on the linear combination of the inputs $v_m = \boldsymbol{w}_m^T \boldsymbol{x} + w_{m0}$, see the bottom panel of Figure 4.1. The hyperbolic tangent

$$\phi(v_m) = \tanh(v_m) = \frac{e^{v_m} - e^{-v_m}}{e^{v_m} + e^{-v_m}} \tag{4.13}$$

is a common nonlinear sigmoidal activation function in the hidden layer. In the output layer, the linear $\varphi(z) = z$ and the softmax activation functions are typical in the regression and classification problems, respectively (Hastie et al., 2001).

The weights of the network α_m and w_{mi}, $m = 1, \ldots, M$, $i = 1, \ldots, d$, and the bias terms α_0 and $\boldsymbol{w}_0$ are optimized based on the training data. Hastie et al. (2001) advice to select a large number of hidden nodes and to use a weight decay regularization

$$\underset{\boldsymbol{\theta}}{\text{minimize}} \quad \frac{1}{N} \sum_{n=1}^{N} (y_n - f(\boldsymbol{x}_n))^2 + \lambda \boldsymbol{\theta}^T \boldsymbol{\theta} , \tag{4.14}$$

where λ is a tuning parameter and the vector $\boldsymbol{\theta}$ includes all the weights of the network. The problem in Equation (4.14) is closely related to the ridge regression used for linear models, see Equation (3.4). The generalization error estimation techniques are used to select a proper value for λ. The inputs are usually standardized to have zero mean and unit variance if a priori knowledge is not available. Scaling ensures that all the inputs are initially equally important.

Since the optimization problem in Equation (4.14) is highly nonlinear, a solution is found by iterative optimization strategies, for instance back-propagation (Haykin, 1999), Levenberg-Marquardt (Bishop, 1995) or scaled conjugate gradient (Møller, 1993) algorithms. The problem is also nonconvex, thus there may exist many local minima. Typically, several different initializations of the weights are used and the solution producing the smallest error is selected.

Another important neural network model is a radial basis function (RBF) network (Broomhead and Lowe, 1988). The feedforward model in the top panel of Figure 4.1 presents also the structure of the traditional RBF network if the bias connections to the hidden nodes are omitted. However, the structure of the hidden node is different from the MLP network. Now, the activation of the hidden node

$$\phi_m(\boldsymbol{x}) = \phi(\|\boldsymbol{x} - \boldsymbol{c}_m\|) \tag{4.15}$$

is defined by the distance between the input $\boldsymbol{x}$ and the center of the basis function $\boldsymbol{c}_m$. The most common basis function is perhaps the Gaussian function

$$\phi_m(\boldsymbol{x}) = \exp\left(\frac{\|\boldsymbol{x} - \boldsymbol{c}_m\|^2}{\sigma_m^2}\right) , \tag{4.16}$$

where σ_m is the scale or width of the basis function and it controls the smoothness of the mapping. Bishop (1995) presents several other choices for the basis function. The output of the RBF network with the Gaussian basis functions is

$$f(\boldsymbol{x}) = \sum_{m=1}^{M} \alpha_m \exp\left(\frac{\|\boldsymbol{x} - \boldsymbol{c}_m\|^2}{\sigma_m^2}\right) + \alpha_0 \ . \tag{4.17}$$

Training of the network can be carried out in two phases (Bishop, 1995). First, the centers $\boldsymbol{c}_m$ and the widths σ_m of a predefined number of basis functions are selected using unsupervised techniques, i.e. based only on the input data. Second, the output layer parameters α_m are optimized by minimizing the error between the network outputs and the observed outputs. In the second phase, the basis functions are fixed and the network corresponds to the nonadaptive model in Equation (4.4) resulting in the evaluation of the parameters α_m by the linear regression techniques. It is also possible to optimize all the parameters $\boldsymbol{c}_m$, σ_m, and $\boldsymbol{\alpha}_m$ in a supervised manner by minimizing the MSE in Equation (4.2). It is a nonlinear and nonconvex problem and a solution can be determined using the same iterative algorithms as in the case of the MLP network.

Alternatively, if the centers of the basis functions are located on the training data points with equal widths $\sigma_m = \sigma$ and the output layer parameters are optimized using ridge regularization as suggested by Orr (1996), the resulting RBF network coincides essentially with the LS-SVM model. Instead of ridge regularization, Bishop (1991) proposes to regularize the error function by the sum of squares of the second order derivatives of the network with respect to the inputs. The regularization should damp out rapid oscillations in the network output.

Also, both input and output variables can be used to determine the placement of the centers. Orr (1995) presents a regularized forward selection (RFS) method that combines the ridge regularization and the forward selection procedure. All the input samples are the candidate centers and the regularization parameter is tuned automatically based on the generalized cross validation criterion, thus only the width σ has to be preset. Peng et al. (2007) propose a continuous forward algorithm (CFA) to train the RBF network. They add the basis functions one at a time into the network. Instead of using fixed values, they consider the width and the center as continuous valued parameters and optimize the values by a conjugate gradient algorithm. Only the most recently added basis function is optimized keeping the rest fixed. The previous approaches do not allow the rejection of already added basis functions, although some of them would become unnecessary in later stages of the algorithms. Huang et al. (2005) introduce a sequential learning strategy called a generalized growing and pruning algorithm for RBF (GGAP-RBF). They define the significance of a basis function by its contribution to the overall performance of the network. A basis function is added or deleted if its significance is greater or smaller than a predefined threshold.

The architectures of both MLP and RBF networks can be represented as a feedforward model and the output is formulated as a linear combination of univariate functions. Since both are universal approximators, there always exists an RBF network that mimics accurately a given MLP network, or vice versa (Haykin, 1999). Nevertheless, there are also important differences between the MLP and RBF networks (Bishop, 1995; Haykin, 1999). The RBF networks include only one hidden layer whereas the MLP network may consists of several ones. The activation of the hidden node is evaluated by the inner product between the input and weight vectors in the MLP network and by the distance between the input and the center of the basis function in the RBF network. In addition, where the sigmoid activation function in the MLP network has global characteristics, the radial basis functions are typically localized.

4.3.1 Sensitivity analysis

In several applications, it is required that the model gives insight into importance of inputs in addition to accurate predictions. Obviously, all the inputs are rarely equally important. Several criteria exist to assess relative importance of the input variables. The value of a criterion for a single input is typically noninformative itself, but the inputs can be ranked by comparing all the values with each other. Leray and Gallinari (1999) distinguish between the zero, first and second order methods.

The zero order methods use only the weights of the network to rank the inputs. Yacoub and Bennani (2000) evaluate the relative contributions based on the absolute values of weights. Both input and output layer weights are considered by evaluating a product between them. It is assumed that important inputs have the large absolute weights.

The first order methods are based on the derivatives of the network. Ruck et al. (1990) propose to measure relevance by the sum of the absolute derivatives

$$S_i = \sum_{n=1}^{N} |\delta_{ni}|, \quad \text{where} \quad \delta_{ni} = \frac{\partial f(\boldsymbol{x}_n)}{\partial x_i} \quad \text{and} \quad i = 1, \ldots, d \ . \tag{4.18}$$

Observe, that the derivatives in Equation (4.18) are not the same that are used in the training of the model, i.e. the derivatives with respect to the weights. Each partial derivative describes the local rate of change in the output with respect to the corresponding input while the rest are kept fixed. The inputs are ranked based on the values of relevance measure S_i and the most important inputs have the largest values. In addition to the ranking of inputs, a set of scatter graphs of the partial derivative δ_{ni} versus the input x_{ni} can be plotted (Gevrey et al., 2003). The graphs enable the visual justification of the dependencies.

In addition to the sum of the absolute derivatives, Refenes and Zapranis (1999) propose other quantities, for instance the sum of the squared derivatives, the standard deviation of derivatives, and weighting the values $|\delta_{ni}|$ by $|x_{ni}/f(\boldsymbol{x}_n)|$, $f(\boldsymbol{x}_n) \neq 0$ in Equation (4.18). Nevertheless, Leray and Gallinari (1999) remark that the effect of extreme observations should be discarded in order to obtain robust methods. In Publication 5, it is proposed to measure relevance by

$$S_i = \gamma m_{\delta_i} + (1-\gamma)\Delta_{\delta_i}, \quad \text{where} \quad \gamma \in [0,1] \ . \tag{4.19}$$

It is the weighted sum of the median m_{δ_i} and the variability Δ_{δ_i} of the absolute values $|\delta_{ni}|$. The variability is defined as a difference $\Delta_{\delta_i} = |\delta_{ni}|^{\text{high}} - |\delta_{ni}|^{\text{low}}$, where $|\delta_{ni}|^{\text{high}}$ and $|\delta_{ni}|^{\text{low}}$ are the $(1-q)N^{\text{th}}$ and qN^{th} values in the ordered list of the N absolute values $|\delta_{ni}|$. The parameter $q \in (0, 0.5)$ defines the width of the central interval of the absolute values. Thus, both the median m_{δ_i} and the difference Δ_{δ_i} are insensitive to the outliers in the data. Either a large variability Δ_{δ_i} or a clearly nonzero median m_{δ_i} may indicate that the output is sensitive to the corresponding input x_i. In Equation (4.19), the parameter γ defines the weighting between the median m_{δ_i} and the variability Δ_{δ_i} in the ranking of inputs, i.e. the ordering of inputs may be changed by varying γ. The choices $\gamma \in [0, 0.5)$ and $\gamma \in (0, 5, 1]$ put more emphasis on the variability and the median, respectively, whereas the choice $\gamma = 0.5$ does not impose a priori preference.

In the second order methods, the evaluation of relevances is based on the second derivatives of the network. Instead of the derivatives with respect to the inputs, Cibas et al. (1996) suggest to apply the partial derivatives with respect to the weights of the network. The relevance of the input x_i is defined as

$$S_i = \frac{1}{2} \sum_{m=1}^{M} \frac{\partial^2 \text{MSE}}{\partial w_{mi}^2} w_{mi}^2 \ , \tag{4.20}$$

where w_{mi} is the connection weight between the i^{th} input and the m^{th} hidden node. This relevance measure approximates the increase in the error function caused by deleting the input x_i. Naturally, the most relevant input causes the largest increase.

4.3.2 Input variable selection for the MLP network

All the relevance measures presented in Section 4.3.1 can be used in the selection of input variables for the MLP network. In practice, the selection is based on backward elimination. The network is trained using all the available inputs and the least relevant input or subset of inputs is deleted. Castellano and Fanelli (2000) show a fast strategy to adjust the remaining weights after the elimination of input. However, they keep the number of nodes in the hidden layer fixed, which may be suboptimal. Better generalization performance may be obtained by repeating the nonlinear optimization of the weights from novel initializations and validating the network architecture. On the other hand, the computational burden quickly gets intolerable with an increasing number of inputs.

The input variables can also be selected indirectly using the weight pruning strategies. Initially, an oversized network that overfits to the data is trained. Subsequently, the network architecture is reduced by eliminating connection weights. The input x_i does not contribute to the network output at all when all the weights w_{mi}, $m = 1, \ldots, M$ are deleted. In the optimal brain damage (OBD) by Le Cun et al. (1990), the minimal increase in the training error is used as the criterion for pruning the weight. The changes in the cost function are determined by the diagonal quadratic approximation, i.e. using only the diagonal elements of the Hessian matrix. Hassibi and Stork (1993) recommend to use the full Hessian matrix by introducing the optimal brain surgeon (OBS). However, the inverse of the Hessian matrix must be evaluated in the OBS, which increases the computational complexity compared to the OBD.

In Publication 4, the input variables are selected by the SISAL method (see Section 3.2.2) and the MLP network is trained using the obtained inputs. The proposed filter strategy is applied to time series prediction problems with different degrees of nonlinearities. It is compared to the MLP networks that are built using the forward selection algorithm and to the MLP networks using all the inputs. Importance of the inputs obtained by SISAL reflect importance of the inputs in the nonlinear models very well. In the MLP networks, relative importance of the inputs are evaluated using the partial derivatives. In summary, SISAL combined with the MLP network offers a reasonable compromise between the prediction accuracy and the computational complexity. Following Li and Peng (2007), linear-in-the-parameter models, e.g. polynomials, could be used in the input selection to improve the performance of SISAL in highly nonlinear problems.

In Equation (4.14), the weight decay regularization encourages the weights to be close to zero. However, it is unlikely that any of the weights would be exactly zero, thus variable selection is not performed. In order to discard an input x_i from the MLP network, it is required that all the outcoming weights $\boldsymbol{w}_i = [w_{1i}, \ldots, w_{Mi}]^T$ from an input x_i are zero. Grandvalet and Canu (1999); Chapados and Bengio (2001); Similä and Tikka (2008) propose techniques to perform simultaneous model fitting and network architecture selection. All approaches penalize the mean squared error cost function by groups of weights instead of the individual weights.

Grandvalet and Canu (1999) present an adaptive ridge regression

$$\begin{aligned} &\underset{\boldsymbol{w}_1,\ldots,\boldsymbol{w}_d,\boldsymbol{\alpha}}{\text{minimize}} && \frac{1}{N}\sum_{n=1}^{N}(y_n - f(\boldsymbol{x}_n))^2 + \sum_{i=1}^{d}\lambda_i\|\boldsymbol{w}_i\|_2^2 + \lambda_{d+1}\sum_{m=1}^{M}\alpha_m^2 \\ &\text{such that} && \frac{1}{d+1}\sum_{i=1}^{d+1}\frac{1}{\lambda_i} = \frac{1}{\lambda}, \quad \lambda_i > 0 \ , \end{aligned} \tag{4.21}$$

where the vector $\boldsymbol{w}_i$ includes all the outcoming weights from the input x_i and α_m, $m = 1, \ldots, M$ are the output layer parameters. Each vector of weights $\boldsymbol{w}_i$ has its own regularization parameter λ_i. The additional constraint on regularization parameters is imposed to decrease the number of hyperparameters from $d + 1$ to one. An additional advantage of the constraint is that with a large enough λ some of the parameters λ_i will be zero and the corresponding inputs are discarded from the model. The regularization of output layer parameters affects smoothness of the mapping. However, in the experiments by Grandvalet and Canu (1999) the adaptive ridge regression is only applied to the additive models and no results for the MLP network are reported.

Chapados and Bengio (2001) propose the input decay

$$\underset{\boldsymbol{w}_1,\ldots,\boldsymbol{w}_d,\boldsymbol{\alpha}}{\text{minimize}} \ \frac{1}{N}\sum_{n=1}^{N}(y_n - f(\boldsymbol{x}_n))^2 + \lambda\sum_{i=1}^{d}\frac{\|\boldsymbol{w}_i\|_2^2}{\eta + \|\boldsymbol{w}_i\|_2^2} \ . \tag{4.22}$$

The regularization term has two purposes. First, it prevents the outcoming weights of the useless inputs being far from zero. Second, on a certain point determined by the hyperparameter η, the regularization term is approximately constant and independent from the values of weights. With small values of $\|\boldsymbol{w}\|_2$, the regularization term is nearly quadratic and it mimics the weight decay. A disadvantage of the input decay is that it does not regularize the output layer weights at all and a proper number of hidden nodes must be validated using model selection techniques.

Similä and Tikka (2008) introduce the following penalized optimization problem for simultaneous input variable and hidden node selection

$$\underset{\boldsymbol{w}_1,\dots,\boldsymbol{w}_d,\boldsymbol{\alpha}}{\text{minimize}} \quad \frac{1}{N}\sum_{n=1}^{N}(y_n - f(\boldsymbol{x}_n))^2 + \lambda\left(\sum_{i=1}^{d} p(\|\boldsymbol{w}_i\|_2) + \sum_{m=1}^{M} p(|\alpha_m|)\right) \quad , \tag{4.23}$$

where $p(\cdot)$ is a penalty function. The maximum number of hidden nodes M is predefined, but only useful inputs and hidden nodes are effective after the training. Similä (2007b) shows that a differentiable penalty function $p(s)$, $s \geq 0$ encourages blockwise sparse solutions only when $p'(0) > 0$ holds. Both weight decay $p(s) = s^2$ and input decay $p(s) = s^2/(\eta+s^2)$ penalties have $p'(0) = 0$ and they are not able to produce sparse solutions in theory. Similä and Tikka (2008) use the function $p(s) = c\log(1 + s/c)$, see also Equation (3.18), that satisfies the sparsity condition since $p'(0) = 1$. The parameter c controls the amount of penalization that is imposed on the weights. By decreasing the value of c, the amount of penalization on large values of $\|\boldsymbol{w}\|_2$ is decreased, thereby it has a similar effect as η in Equation (4.22). Since the penalty function is nondifferentiable in the origin, the problem cannot be solved directly by gradient methods. Similä and Tikka (2008) use a logarithmic barrier reformulation to approximate the original problem and propose an efficient optimization method to find a solution.

4.3.3 Input variable selection for the RBF network

The partial derivatives of the network with respect to the input variables can also be used in variable selection for the RBF network. The weight pruning strategies, such as the OBD and the OBS, cannot reduce the number of input variables in the standard RBF network in Equation (4.17), since each input does not have its own weights. However, usage of the adaptive weights in the basis functions as in Equation (4.11) enables input selection by weight pruning. Also, the input selection approaches for the SVM that are based on the adaptive weights could be modified to be applicable for the RBF networks as well. Villmann et al. (2006) compare several metric adaptation methods and report that the scaled Euclidean metric produces competitive results. In addition, they notify that problem adapted metrics are generally superior to the standard Euclidean distance.

Variable selection can also be incorporated into the two-phase training strategy of the RBF networks. Bishop (1995) suggests to consider setting of the basis function parameters as a density estimation problem. The input data distribution is estimated by the Gaussian mixture model. Furthermore, means and variances of the Gaussian distributions are used as the centers $\boldsymbol{c}_m$ and widths σ_m of the basis functions. Dy and Brodley (2004) apply the forward selection algorithm to find relevant input variables for the mixture model. They also show that the maximum likelihood solution is biased toward lower dimensions and they propose a normalization scheme for the selection criterion. The number of component distributions is automatically determined for each subset of inputs, which results in better performance than usage of a predefined number of components. Law et al. (2004) avoid combinatorial search by casting the selection of inputs as an estimation problem. For each input, a real valued saliency $s_i \in [0, 1]$ is estimated. The saliencies of the irrelevant inputs should be zero. Eventually, they combine the determination of number of component distributions with the estimation of saliencies. The main disadvantage of unsupervised techniques is that the output observations are not considered in variable selection at all. A practical subset of inputs for the clustering task may be unsatisfactory for the prediction purposes.

Sequential input selection algorithm for the RBF network

In Publication 5, a sequential input selection algorithm for the radial basis function (SISAL-RBF) network is presented. The algorithm starts by training the RBF network using all the input variables. The selection of centers $\boldsymbol{c}_m$ is circumvented by placing a Gaussian basis function on each training data point. Equal predefined widths $\sigma_m = \sigma$ are applied and the output layer parameters α_m are estimated using the ridge regularization. The training procedure includes two hyperparameters, i.e. the width σ and the regularization parameter λ. However, it is possible to tune λ automatically as it is explained below.

Orr (1996) shows that for fixed values of σ and λ, the leave-one-out CV error can be evaluated analytically without retraining and testing the network N times. Furthermore, keeping σ fixed the leave-one-out CV error can be written as a function of λ as follows

$$\mathrm{MSE}_{\mathrm{LOO}}(\lambda) = \frac{1}{N}\boldsymbol{y}^T \boldsymbol{P} \left(\mathrm{diag}(\boldsymbol{P})\right)^{-2} \boldsymbol{P}\boldsymbol{y} \ , \tag{4.24}$$

where $\boldsymbol{P} = \boldsymbol{I}_N - \boldsymbol{\Phi}(\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \lambda\boldsymbol{I}_M)^{-1}\boldsymbol{\Phi}^T$ and diag($\boldsymbol{P}$) is of the same size and has the same diagonal as $\boldsymbol{P}$ but is zero off the diagonal. The elements of matrix $\boldsymbol{\Phi}$ are defined as $\boldsymbol{\Phi}_{nm} = \{\phi_m(\boldsymbol{x}_n)\}$, $n = 1, \ldots, N$, $m = 1, \ldots, M$ and the vector $\boldsymbol{y}$ includes the output observations $\boldsymbol{y} = [y_1, \ldots, y_N]^T$. In Publication 5, the optimal λ is found by minimizing the leave-one-out CV error in Equation (4.24) by the golden section line search method (Bazaraa et al., 1979). Thus, the training methodology has only one preset parameter, i.e. the width σ.

After the training of the network, the input variables are ranked by measuring the relevances using a fixed γ in Equation (4.19). The input producing the smallest value S_i is rejected from the model and the network is retrained using only the remaining inputs. Backward elimination of the inputs produces d nested subsets, where d is the number of inputs. Due to the efficient evaluation of the leave-one-out CV error, it is computationally feasible to optimize the hyperparameters for each subset and to select the inputs based on the validation error instead of the training error. The parameter γ could also be selected based on the validation error. However, it would increase the computational complexity significantly, since it is highly probable that more than d subsets of the inputs needs to be evaluated.

In the experiments of Publication 5, the SISAL-RBF networks have smaller prediction errors than the ordinary RBF networks, i.e. the networks with all the inputs. The forward selection (FS) algorithm and the proposed training strategy perform similarly in terms of both prediction accuracy and variable selection. Table 4.1

Table 4.1: The relative empirical computational times of the RBF networks on data sets with different numbers of input variables d in the experiments of Publication 5.

Data set	d	Ordinary RBF	SISAL-RBF	RBF with FS
Add10	10	1	9.4	47
Boston Housing	13	1	14	93
Bank	32	1	33	540
Wine	256	1	250	30000
Computational complexity		$\mathcal{O}(1)$	$\mathcal{O}(d)$	$\mathcal{O}(d^2)$

shows the relative computational times of the evaluated networks. In the case of SISAL-RBF, a fixed value $\gamma = 0.5$ was applied in the relevance measure, see Equation (4.19). The proposed variable selection algorithm is clearly faster than the FS method and the difference is more and more apparent with a large number of input variables. Table 4.1 also shows that the relative computational complexity of the SISAL-RBF network depends roughly linearly on the number of input variables, whereas the relative dependency of the FS procedure is approximately quadratic. However, the FS algorithm could be stopped before using all the inputs, but already finding the first input requires the evaluation of d networks, which equals to the total number of networks evaluated by SISAL-RBF. The actual running time of SISAL-RBF is more complex than linear with respect to the number of inputs. It also strongly depends on the number of available observations N.

Input selection for the RBF network by constrained optimization

Poggio and Girosi (1990) propose to extend the norm $\|\boldsymbol{x}-\boldsymbol{c}_m\|^2$ in Equation (4.16) by a weighted norm $\|\boldsymbol{x}-\boldsymbol{c}_m\|^2_{\boldsymbol{W}} = (\boldsymbol{x}-\boldsymbol{c}_m)^T\boldsymbol{W}^T\boldsymbol{W}(\boldsymbol{x}-\boldsymbol{c}_m)$, where $\boldsymbol{W}$ is a square matrix of size $d \times d$. However, the number of adaptive weights $\boldsymbol{W} = \{w_{ij}\}_{i,j=1,\ldots,d}$ depends quadratically on the number of input variables. As a compromise between adaptivity and the number of tunable parameters, separate weights can be assigned for the input variables as in Equation (4.11) which corresponds to the diagonal matrix $\boldsymbol{W}$. In Publications 6 and 7, instead of the standard Euclidean norm the adaptive weights are applied in the Gaussian basis functions. The output of an adaptively weighted radial basis function (AW-RBF) network is

$$f(\boldsymbol{x}, \boldsymbol{w}, \boldsymbol{\alpha}) = \sum_{m=1}^{M} \alpha_m K_{\boldsymbol{w}}(\boldsymbol{c}_m, \boldsymbol{x}) + \alpha_0 \tag{4.25}$$

and $K_{\boldsymbol{w}}(\boldsymbol{c}_m, \boldsymbol{x})$ is defined in Equation (4.11). A large weight w_i, $i = 1, \ldots, d$ corresponds to a relevant input and the elimination of input variable occurs by setting $w_i = 0$.

Poggio and Girosi (1990) suggest to optimize the weight matrix $\boldsymbol{W}$, the centers $\boldsymbol{c}_m$, and the output layer parameters $\boldsymbol{\alpha}$ by minimizing a regularized cost function using a gradient descent algorithm. However, no explicit constraints are imposed on the weight matrix $\boldsymbol{W}$. In Publication 6, the parameters of the AW-RBF network are determined by solving the following constrained optimization problem

$$\begin{aligned} &\underset{\boldsymbol{w},\boldsymbol{\alpha}}{\text{minimize}} \quad E_\lambda(\boldsymbol{w},\boldsymbol{\alpha}) = \frac{1}{N}\sum_{n=1}^{N}(y_n - f(\boldsymbol{x}_n,\boldsymbol{w},\boldsymbol{\alpha}))^2 + \lambda\sum_{m=1}^{M}\alpha_m^2 \\ &\text{such that} \quad \sum_{i=1}^{d} w_i \leq t \quad \text{and} \quad w_i \geq 0, \quad i = 1,\ldots,d \ , \end{aligned} \tag{4.26}$$

where λ and t are the regularization and shrinkage parameters, respectively. The ridge regularization is applied to the output layer parameters $\boldsymbol{\alpha}$, since a basis function is placed on each training data point. The constraint $\sum_{i=1}^{d} w_i \leq t$ shrinks the values of weights toward zero. With a small enough t, some of the weights are exactly zero. The nonnegativity constraints $w_i \geq 0$ guarantee that the basis function in Equation (4.11) is localized. Grandvalet and Canu (1999) apply the similar constraints on the diagonal weighting matrix in training of the SVM classifier.

In Publication 6, a logarithmic barrier function method (Bazaraa et al., 1979; Boyd and Vandenberghe, 2004) is used to approximate the constrained problem in Equation (4.26) by an unconstrained one. A solution is found by performing successive minimizations with respect to the output layer parameters $\boldsymbol{\alpha}$ and to the adaptive weights $\boldsymbol{w}$, that is, when the parameters $\boldsymbol{\alpha}$ are optimized the weights $\boldsymbol{w}$ are fixed and vice versa. The output layer parameters $\boldsymbol{\alpha}$ are updated by solving a system of linear equations, whereas updating of the adaptive weights $\boldsymbol{w}$ is carried out by an iterative optimization algorithm. A convergence analysis of the alternating optimization strategy is provided by Bezdek et al. (1987).

An AW-RBF network obtained by solving the problem in Equation (4.26) is not sparse in terms of the basis functions due to the ridge regularization. To be able to select also the basis functions, a modified version of the problem is considered in Publication 7. The LASSO type constraints are imposed on both output layer parameters and weights leading to the following constrained optimization problem

$$\begin{aligned} &\underset{\boldsymbol{w},\boldsymbol{\alpha}}{\text{minimize}} \quad E(\boldsymbol{w},\boldsymbol{\alpha}) = \frac{1}{N}\sum_{n=1}^{N}(y_n - f(\boldsymbol{x}_n,\boldsymbol{w},\boldsymbol{\alpha}))^2 \\ &\text{such that} \quad \sum_{m=0}^{M}|\alpha_m| \leq r \ , \\ &\qquad\qquad \sum_{i=1}^{d} w_i \leq t, \quad \text{and} \quad w_i \geq 0, \quad i = 1,\ldots,d \ . \end{aligned} \tag{4.27}$$

The basis functions are initially located on each training data point. With an appropriate choice of the shrinkage parameter r, some of the output layer parameters are shrunk toward zero and some of them are set exactly to zero corresponding to the rejection of basis functions. The values of adaptive weights are penalized similarly as in Equation (4.26) to encourage sparsity also in terms of the inputs.

Due to the absolute values, the constraint on the output layer parameters α_m, $m = 1,\ldots,M$, is nondifferentiable when at least one of the parameters α_m equals to zero. For fixed values of the weights $\boldsymbol{w}$, the output layer parameters $\boldsymbol{\alpha}$ cannot be updated as easily as in the case of the ridge regularization although efficient algorithms to solve the LASSO problem exist (Schmidt et al., 2007). In Publication 7, the problem in Equation (4.27) is approximated by a logarithmic barrier reformulation that is differentiable with respect to both $\boldsymbol{\alpha}$ and $\boldsymbol{w}$, thus, a direct optimization approach is proposed in addition to the alternating optimization strategy. The direct approach means simultaneous optimization of $\boldsymbol{\alpha}$ and $\boldsymbol{w}$ and the proposed method is inspired by the Levenberg-Marquardt optimization algorithm. In the experiments of Publication 7, the alternating and direct approaches produce similar results.

The problems in Equations (4.26) and (4.27) contain two continuous valued hyperparameters (λ, t) and (r, t), respectively. The hyperparameters control the tradeoff between the prediction accuracy and complexity of the resulting network. The model selection is carried out by minimizing an estimate of the generalization error using a predefined two-dimensional grid of the hyperparameters. For comparison, the SVM with the Gaussian kernel in the regression problem in Equation (4.7) contains three tuning parameters (ϵ, σ, C). However, the SVM retains the original number of input variables, whereas the AW-RBF network is able to discard useless inputs as shown in the experiments of Publications 6 and 7. In addition to input variable selection, the weights w_i, $i = 1, \ldots, d$ describe relative importance of the inputs in the network, such that the most relevant inputs have the largest weights. To make the weights comparable with each other, the inputs should have similar scales. This is obtained easily by standardizing the inputs to have zero mean and unit variance before the training. Moreover, no computationally expensive search strategies are needed for input and basis function selection.

An illustrative example

Bishop (1995) reports that the RBF networks are sensitive to the input variables which have significant variance but which are not important in the determination of the output variable. In such a case, the RBF network has a large error even if the number of basis functions is increased significantly. As an example, let us consider a data set $\{\boldsymbol{x}_n, y_n\}_{n=1}^{N}$, where the inputs $\boldsymbol{x}_n = [x_{n1}, x_{n2}]$ are sampled independently from the uniform distribution $x_{ni} \sim \mathcal{U}(-3, 3)$. The output y_n is generated from the model

$$y_n = \mathrm{sinc}(x_{n1}) + \varepsilon_n, \quad n = 1, \ldots, N \ , \tag{4.28}$$

where independently normally distributed samples $\varepsilon_n \sim \mathrm{N}(0, 0.15^2)$ are used as the additive noise. The sizes of the training and validation sets are $N_t = 100$ and $N_v = 2000$, respectively.

The objective is to compare the standard RBF network with the AW-RBF network trained by solving the problem in Equation (4.27). In both networks, a basis function is initially located on each training data point. The standard network is trained using the same fixed width σ in all the Gaussian basis functions and applying the ridge regularization to restrict the values of the output layer parameters. The training is repeated using 20 values of the regularization parameter λ and the width σ, which are logarithmically equally spaced in the ranges $\lambda \in [10^{-3}, 5]$ and $\sigma^2 \in [0.02, 10]$. The AW-RBF network is trained using 20 logarithmically equally spaced points of the shrinkage parameters r and t ($r \in [1, 200]$ and $t \in [0.1, 100]$).

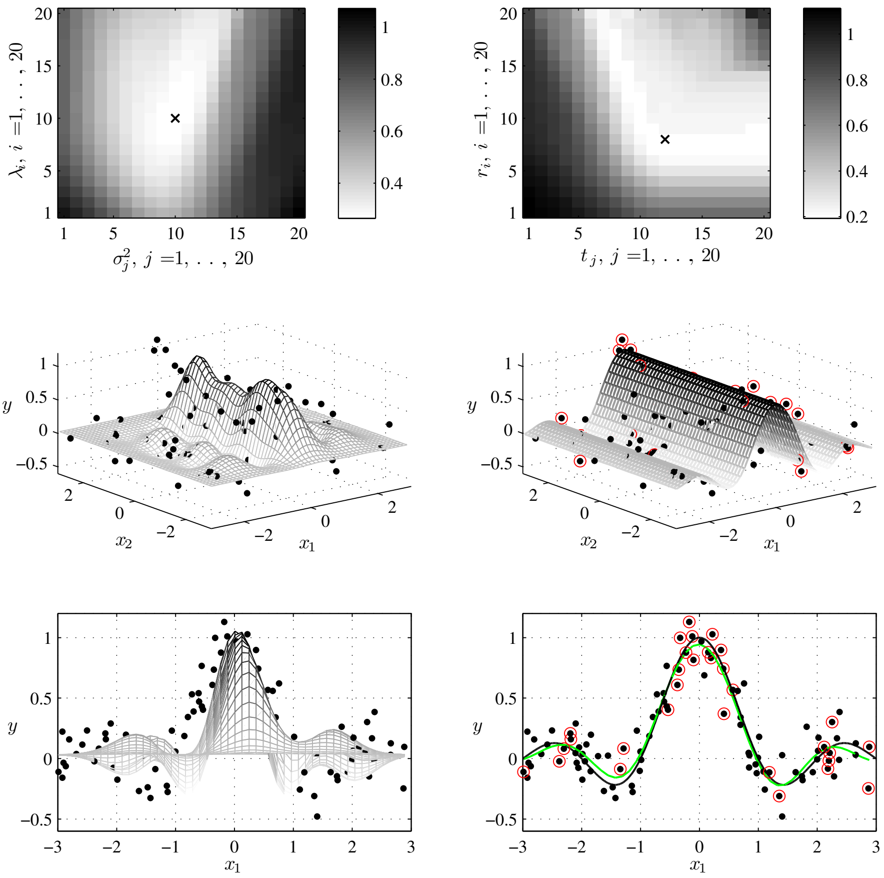

Figure 4.2: The normalized mean squared validation errors (*top*), the estimated output of the minimum validation error models as a function of x_1 and x_2 (*middle*) and the projections on x_1 axis (*bottom*). The left and right panels correspond to the standard RBF network and to the AW-RBF network trained by the constrained problem in Equation (4.27), respectively. The black crosses mark the minimum validation errors, the black dots are the training samples and the selected basis functions are marked by the red circles.

The results are summarized in Figure 4.2. The validation errors of both networks (*the top row*) show that the network outputs depend evidently on both of their hyperparameters and the minima are found from the centers of the grids indicated by the black crosses. However, observe that smaller minimum is obtained by the AW-RBF network than by the standard network. In the original scale, the minimum validation MSE of the AW-RBF network is 0.025 which is an accurate estimate for the known variance of the noise 0.15^2. The middle and bottom rows show that the irrelevant input x_2 does not affect the output of AW-RBF network (*the right panel*), whereas the output of the standard RBF network depends strongly on it (*the left panel*). In addition, the standard network has clearly difficulties to model

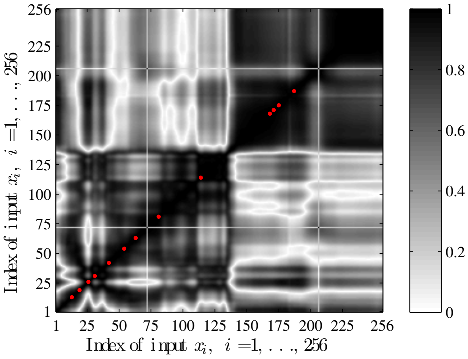

Figure 4.3: The absolute correlations between the input variables x_i, $i = 1, \ldots, 256$ in Wine data. The red dots mark the inputs that are selected in the experiments of Publication 7.

the borders of the input space. The bottom right panel shows that the AW-RBF network (*the green line*) approximates nearly perfectly the actual sinc function (*the black line*). Only 31 out of the initial 100 basis functions are included into the AW-RBF network that produces the minimum validation error.

Stability of the AW-RBF network

Let us assume that a model $\mathcal{M}$ is trained using the data $\mathcal{D}$. According to Breiman (1996), a training procedure is stable if a small change in the data $\mathcal{D}$ does not cause large changes in the resulting model $\mathcal{M}$. Stability of the AW-RBF network, that is trained by minimizing the constrained problem in Equation (4.27), is illustrated by revisiting the experiment on Wine data in Publication 7. The purpose is to predict the level of alcohol of wine from its observed mean infrared spectrum. The data include $d = 256$ input variables and $N_{\text{tr}} = 94$ and $N_{\text{test}} = 30$ observations in the training and test sets, respectively. The absolute correlations between the input variables are shown in Figure 4.3. The very high absolute correlations imply that several inputs contain nearly identical information which complicates variable selection substantially. In the experiments of Publication 7, only one input is typically chosen from the groups of highly correlated inputs, see Figure 4.3.

The bootstrap (Efron and Tibshirani, 1993) is a convenient resampling strategy to generate modified training data sets. A bootstrap sample is created by resampling the original training data $\{\boldsymbol{x}_n, y_n\}_{n=1}^{N_{\text{tr}}}$ with replacement. The size of the bootstrap sample equals to the size of the original data N_{tr} but some of the pairs $\{\boldsymbol{x}_n, y_n\}$ appear zero times, once, or twice, etc. in the sample. The resampling procedure and the training of the AW-RBF network are repeated $B = 500$ times. With each resampled data set, the same values of the shrinkage parameters r and t are applied. These values produced the minimum validation error in the analysis of the original data in Publication 7. The results are summarized in Figure 4.4. Averages with standard deviations in parentheses of 500 replications of the test error and the number of selected inputs are 0.0053(0.0015) and 12.4(2.0), respectively. The

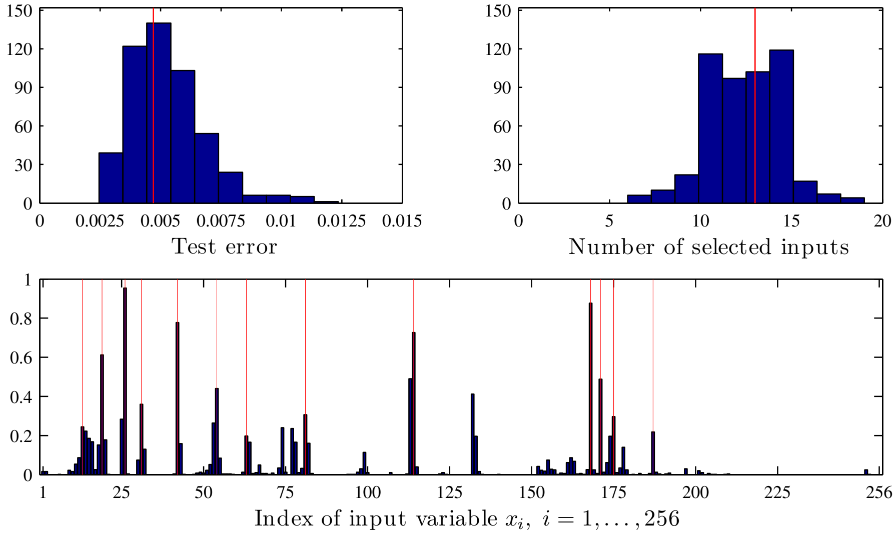

Figure 4.4: Empirical distributions of the test error (*the top left panel*) and the number of selected inputs (*the top right panel*) for the AW-RBF network evaluated using the bootstrap resampling procedure. The frequency of the inclusion of each input variable to the AW-RBF network in the bootstrap replications (*the bottom panel*). The red vertical lines show the test error, the number of inputs and the selected inputs in the AW-RBF network, when the original data is used in the training.

relatively small standard deviations indicate that the replications are concentrated close to the average values, which is also shown in the top panel of Figure 4.4. The averages correspond also very well to the test error 0.0047 and to the number of inputs 13 that are obtained using the original data. The bottom panel of Figure 4.4 presents the frequencies of the appearances of each input variable in the bootstrap replications. In many replications, several of the chosen variables are the same that are also found with the original data. In the case that the chosen input does not coincide exactly with one of the 13 originally selected inputs, it is typically highly correlated with at least one of them, see also Figure 4.3. In addition, it is notable that most of the inputs are always excluded from the AW-RBF network.

Chapter 5

Summary and conclusions

Efficient computational data analysis techniques are of the utmost importance due to the growth of sizes of data sets. For instance, Guyon and Elisseeff (2003) note that the number of variables has increased from dozens to tens or hundreds of thousands in a short period of time. Data analysis methods should discover relationships that are useful and improve knowledge of the system. However, a model producing accurate predictions but including thousands of variables is not fully satisfactory, since interpretation of dependencies between the variables may be difficult. According to Fayyad et al. (1996), an understanding of the model and its induction makes the model more attractive for practical problems.

In the present thesis, several input variable selection methods are proposed to improve the interpretability of prediction models in both linear and nonlinear regression problems. By the improved interpretability it is meant that a model including only a parsimonious set of inputs points out important dependencies better than a model including all the inputs. The sparsity is also a reasonable priori assumption, since connections are often sparse in the real world, that is, typically any given variable only depends on a relatively small subset of others. In addition, if sparse and full models have comparable prediction errors, the simpler model should be preferred based on the sparsity principle (Blumer et al., 1987). Thus, it is natural to define usefulness of an input variable according to its prediction capability.

In Publication 4, SISAL is introduced as a preprocessing step for the training of the MLP network. However, SISAL can also be applied to variable selection for a single response linear regression model. It performs favorably in comparison with the forward selection and LARS algorithms as presented in Section 3.3.4. The results show that SISAL produces accurate and sparse models also in the case of correlated inputs.

The simultaneous estimation of multiple responses using the same subset of input variables is considered in the context of linear models. In Publications 1 and 2, the MRSR algorithm is presented. It is an extension of the LARS algorithm to the case of multiple response variables. The inputs are added one by one to the model, and the order of additions reflects importance of the inputs. The main advantages of the MRSR algorithm are that it is straightforward to implement and it is computationally efficient also in high-dimensional spaces. Instead of the stepwise search, simultaneous selection and shrinkage of the inputs is performed by minimizing a constrained optimization problem in Publication 3. The efficient evaluation of all the solutions as a function of the shrinkage parameter is introduced. Similä and Tikka (2008) utilize a similar constraint structure in the training of the MLP network resulting in sparse networks in terms of both input variables and hidden nodes.

Nonlinear regression techniques include hyperparameters that control smoothness of the resulting mapping. Nevertheless, the hyperparameters typically restrict only the effective complexity of the model without reducing the original number of input variables. SISAL-RBF is proposed in Publication 5. It is competitive for input variable selection for two reasons. First, it includes only one hyperparameter, the width of the basis function, since the regularization parameter is determined automatically using the leave-one-out CV error. Second, the backward elimination of inputs depends linearly on the number of input variables as opposed to usual quadratic dependency. In Publications 6 and 7, the idea of the usage of a weighted norm instead of the standard Euclidean distance in the Gaussian basis function by Poggio and Girosi (1990) is adopted and developed further. The constraint optimization problems are proposed for simultaneous scaling and selection of the inputs. Depending on the structure of the constraint imposed on output layer parameters, either the effective or actual number of basis functions is reduced. In both cases, variable selection and complexity control is performed by minimizing a single optimization problem, that includes two continuous-valued hyperparameters.

Extensive experiments are conducted to empirically evaluate the proposed variable selection methods. They are compared to numerous other methods using both artificial and real world benchmark data sets. The complexity of each model, such as the number of inputs in the stepwise methods or the hyperparameters in the optimization problems, is always automatically defined according to an estimate of the generalization error. Eventually, the models are compared and assessed based on the results on a test data set that is completely independent from the data sets that are used in the training and the complexity determination of the models. All the conclusions, for example the usefulness of a subset of inputs, are supported by the prediction accuracy of the model on the test data.

The following arguments justify the proposed input selection methods. First, the sparse models have nearly always smaller prediction errors than the models including all the input variables. Second, the suggested methods are at least equally accurate as the other evaluated variable selection methods. Third, in most of the experiments the proposed algorithms select smaller numbers of inputs than the competing approaches. Fourth, sparsity places focus on the useful variables and makes interpretation of the model easier. Fifth, relative importance of the selected inputs is assessed. Sixth, the training is computationally efficient. Seventh, the correct inputs are found in the cases of artificial data sets. The ultimately best subsets for real benchmark data sets are not known, but the obtained subsets certainly produce competitive prediction results.

There are obviously plenty of possibilities for future research. Overall, it would be extremely insightful to validate the interpretability of the proposed methods in cooperation with domain experts. In several applications, the number of inputs exceeds the number of observations. In such a case, SISAL is not directly applicable, but it could be combined with the ridge regularization. Furthermore, the additional hyperparameter could be determined similarly as in SISAL-RBF. The applicability of the proposed methods would also be improved by extensions to classification problems. A straightforward reformulation for the AW-RBF network is to replace the current cost function with the negative log-likelihood function that is shown in Equation (2.12). Moreover, Girolami (2002) and Dhillon et al. (2004) apply kernel functions to clustering of data. Perhaps, an adaptively weighted kernel could be utilized in that context as well.

Bibliography

Abraham, B. and Merola, G. (2005). Dimensionality reduction approach to multivariate prediction, *Computational Statistics & Data Analysis* **48**(1): 5–16.

Akaike, H. (1974). A new look at the statistical model identification, *IEEE Transactions on Automatic Control* **AC–19**(6): 716–723.

Antoniadis, A., Gregoire, G. and McKeague, I. W. (1994). Wavelet methods for curve estimation, *Journal of the American Statistical Association* **89**(428): 1340–1353.

Atkeson, C. G., Moore, A. W. and Schaal, S. (1997). Locally weighted learning, *Artificial Intelligence Review* **11**(1–5): 11–73.

Bakin, S. (1999). *Adaptive regression and model selection in data mining problems*, PhD thesis, The Australian National University, School of Mathematical Sciences, Canberra, Australia.

Baldi, P. and Hornik, K. (1989). Neural networks and principal component analysis: Learning from examples without local minima, *Neural Networks* **2**(1): 53–58.

Baldi, P. and Hornik, K. (1995). Learning in linear neural networks: A survey, *IEEE Transactions on Neural Networks* **6**(4): 837–858.

Barrett, B. E. and Gray, J. B. (1994). A computational framework for variable selection in multivariate regression, *Statistics and Computing* **4**(3): 203–212.

Baudat, G. and Anouar, F. (2001). Kernel-based methods and function approximation, *Proceedings of the International Joint Conference on Neural Networks (IJCNN 2001)*, Vol. 2, IEEE, pp. 1244–1249.

Bazaraa, M. S., Sherali, H. D. and Shetty, C. M. (1979). *Nonlinear programming theory and applications*, John Wiley & Sons, Inc.

Ben-Hur, A., Horn, D., Siegelmann, H. T. and Vapnik, V. (2001). Support vector clustering, *Journal of Machine Learning Research* **2**: 125–137.

Benítez, J. M., Castro, J. L. and Requena, I. (1997). Are artificial neural networks black boxes?, *IEEE Transactions on Neural Networks* **8**(5): 1156–1164.

Bezdek, J. C., Hathaway, R. J., Howard, R. E., Wilson, C. A. and Windham, M. P. (1987). Local convergence analysis of grouped variable version of coordinate descent, *Journal of Optimization Theory and Applications* **54**(3): 471–477.

Bezdek, J. C. and Pal, N. R. (1998). Some new indexes of cluster validity, *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics* **28**(3): 301–315.

Bi, J., Bennett, K. P., Embrechts, M., Breneman, C. M. and Song, M. (2003). Dimensionality reduction via sparse support vector machines, *Journal of Machine Learning Research* **3**: 1229–1243.

Bishop, C. (1991). Improving the generalization properties of radial basis function neural networks, *Neural Computation* **3**(4): 579–588.

Bishop, C. M. (1995). *Neural networks for pattern recognition*, Oxford University Press.

Blum, A. L. and Langley, P. (1997). Selection of relevant features and examples in machine learning, *Artificial Intelligence* **97**(1–2): 245–271.

Blumer, A., Ehrenfeucht, A., Haussler, D. and Warmuth, M. K. (1987). Occam's razor, *Information Processing Letters* **24**(6): 377–380.

Boser, B. E., Guyon, I. M. and Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers, *Proceedings of the Fifth Annual Workshop on Computational Learning Theory (COLT 1992)*, ACM, pp. 144–152.

Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*, Cambridge University Press.

Breiman, L. (1995). Better subset regression using the nonnegative garrotte, *Technometrics* **37**(4): 373–384.

Breiman, L. (1996). Heuristics of instability and stabilization in model selection, *Annals of Statistics* **24**(6): 2350–2383.

Breiman, L. (2001). Statistical modeling: The two cultures, *Statistical Science* **16**(3): 199–231.

Breiman, L. and Friedman, J. H. (1997). Predicting multivariate responses in multiple linear regression, *Journal of the Royal Statistical Society. Series B (Methodological)* **59**(1): 3–54.

Broomhead, D. S. and Lowe, D. (1988). Multivariate functional interpolation and adptive networks, *Complex Systems* **2**(3): 321–355.

Brown, P. J. and Zidek, J. V. (1980). Adaptive multivariate ridge regression, *The Annals of Statistics* **8**(1): 64–74.

Burges, C. J. C. (1998). A tutorial on support vector machines for pattern recognition, *Data Mining and Knowledge Discovery* **2**(2): 121–167.

Burnham, A. J., Viveros, R. and MacGregor, J. F. (1996). Frameworks for latent variable multivariate regression, *Journal of Chemometrics* **10**(1): 31–45.

Carroll, J. D. and Arabie, P. (1980). Multidimensional scaling, *Annual Review of Psychology* **31**: 607–649.

Castellano, G. and Fanelli, A. M. (2000). Variable selection using neural-network models, *Neurocomputing 31* **31**(1–4): 1–13.

Cawley, G. C. and Talbot, N. L. C. (2002). Reduced rank kernel ridge regression, *Neural Processing Letters* **16**(3): 293–302.

Chakrabarti, S. and Jeyasurya, B. (2007). An enhanced radial basis function network for voltage stability monitoring considering multiple contingencies, *Electric Power Systems Research* **77**(7): 780–787.

Chapados, N. and Bengio, Y. (2001). Input decay: Simple and effective soft variable selection, *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN 2001)*, Vol. 2, pp. 1233–1237.

Chapelle, O., Vapnik, V., Bousquet, O. and Mukherjee, S. (2002). Choosing multiple parameters for support vector machines, *Machine Learning* **46**(1–3): 131–159.

Chatfield, C. (1995). *Problem solving: A statistician's guide*, Chapman & Hall.

Chatfield, C. (2001). *Time-series forecasting*, Chapman & Hall/CRC.

Cherkassky, V. and Mulier, F. (1998). *Lerning from data*, John Wiley & Sons, Inc.

Cibas, T., Fogelman Soulie, F., Gallinari, P. and Raudys, S. (1996). Variable selection with neural networks, *Neurocomputing* **12**(1–3): 223–248.

Cotter, S. F., Rao, B. D., Engan, K. and Kreutz-Delgado, K. (2005). Sparse solutions to linear inverse problems with multiple measurement vectors, *IEEE Transactions on Signal Processing* **53**(7): 2477–2488.

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

Csiszár, I. and Tusnády, G. (1984). Information geometry and alternating minimization procedures, *Statistics & Decisions, Supplement Issue No. 1,* pp. 205–237.

Cumming, J. A. and Wooff, D. A. (2007). Dimension reduction via principal variables, *Computational Statistics & Data Analysis* **52**(1): 550–565.

Dayhoff, J. E. and DeLeo, J. M. (2001). Artificial neural networks: Opening the black box, *Cancer* **91**(S8).

DeMers, D. and Cottrell, G. (1993). Non-linear dimensionality reduction, *in* S. J. Hanson, J. D. Cowan and C. L. Giles (eds), *Advances in Neural Information Processing Systems 5*, Morgan Kaufmann, pp. 580–587.

Dempster, A. P., Laird, N. M. and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm, *Journal of the Royal Statistical Society, Series B (Methodological)* **39**(1): 1–38.

Dhillon, I. S., Guan, Y. and Kulis, B. (2004). Kernel k-means, spectral clustering and normalized cuts, *in* R. Kohavi, J. Gehrke, W. DuMouchel and J. Ghosh (eds), *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 551–556.

Donoho, D. L. and Johnstone, I. M. (1994). Ideal spatial adaptation by wavelet shrinkage, *Biometrika* **81**(3): 425–455.

Draper, N. R. and Smith, H. (1981). *Applied regression analysis*, John Wiley & Sons, Inc.

Drucker, H., Burges, C. J. C., Kaufman, L., Smola, A. and Vapnik, V. (1997). Support vector regression machines, *in* M. C. Mozer, M. I. Jordan and T. Petsche (eds), *Advances in Neural Information Processing Systems 9*, The MIT Press, pp. 155–161.

Dy, J. G. and Brodley, C. E. (2004). Feature selection for unsupervised learning, *Journal of Machine Learning Research* **5**: 845–889.

Efron, B. (1982). *The Jackknife, the bootstrap and other resampling plans*, Society for Industrial and Applied Mathematics (SIAM).

Efron, B., Hastie, T., Johnstone, I. and Tibshirani, R. (2004). Least angle regression, *The Annals of Statistics* **32**(2): 407–499.

Efron, B. and Tibshirani, R. J. (1993). *An introduction to the bootstrap*, Chapman & Hall.

Espinoza, M., Suykens, J. A. K. and De Moor, B. (2006). Fixed-size least squares support vector machines: a large scale application in electrical load forecasting, *Computational Management Science* **3**(2): 113–129.

Everitt, B. S. and Dunn, G. (1991). *Applied multivariate analysis*, John Wiley & Sons.

Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties, *Journal of the American Statistical Association* **96**(456): 1348–1360.

Fayyad, U., Piatetsky-Shapiro, G. and Smyth, P. (1996). From data mining to knowledge discovery in databases, *AI Magazine* **17**(3): 37–54.

Feelders, A., Daniels, H. and Holsheimer, M. (2000). Methodological and practical aspects of data mining, *Information & Management* **37**: 271–281.

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems, *Annals of Eugenics* **7**: 179–188.

Flake, G. W. and Lawrence, S. (2002). Efficient SVM regression training with SMO, *Machine Learning* **46**(1–3): 271–290.

François, D., Rossi, F., Wertz, V. and Verleysen, M. (2007). Resampling methods for parameter-free and robust feature selection with mutual information, *Neurocomputing* **70**(7–9): 1276–1288.

Friedman, J. H. (1991). Multivariate adaptive regression splines, *The Annals of Statistics* **19**(1): 1–67.

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

George, E. I. (2000). The variable selection problem, *Journal of the American Statistical Association* **95**(452): 1304–1308.

Gevrey, M., Dimopoulos, I. and Lek, S. (2003). Review and comparison of methods to study the contribution of variables in artificial neural network models, *Ecological Modelling* **160**(3).

Girolami, M. (2002). Mercer kernel-based clustering in feature space, *IEEE Transactions on Neural Networks* **13**(3): 780–784.

Girosi, F., Jones, M. and Poggio, T. (1995). Regularization theory and neural networks architectures, *Neural Computation* **7**(2): 219–269.

Golbabai, A. and Seifollahi, S. (2007). Radial basis function networks in the numerical solution of linear integro-differential equations, *Applied Mathematics and Computation* **188**(1): 427–432.

Grandvalet, Y. and Canu, S. (1999). Outcomes of the equivalence of adaptive ridge with least absoplute shrinkage, *in* M. S. Kearns, S. A. Solla and D. A. Cohn (eds), *Advances in Neural Information Processing Systems 11*, The MIT Press, pp. 445–451.

Grandvalet, Y. and Canu, S. (2003). Adaptive scaling for feature selection in SVMs, *in* S. Becker, S. Thrun and K. Obermayer (eds), *Advances in Neural Information Processing Systems 15*, The MIT Press, pp. 553–560.

Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection, *Journal of Machine Learning Research* **3**: 1157–1182.

Guyon, I., Weston, J., Barnhill, S. and Vapnik, V. (2002). Gene selection for cancer classification using support vector machines, *Machine Learning* **46**(1–3): 389–422.

Hand, D., Mannila, H. and Smyth, P. (2001). *Principles of data mining*, The MIT Press.

Hansen, P. and Jaumard, B. (1997). Cluster analysis and mathematical programming, *Mathematical Programming* **79**(1–3): 191–215.

Hartman, E. J., Keeler, J. D. and Kowalski, J. M. (1990). Layered neural networks with Gaussian hidden units as universal approximations, *Neural Computation* **2**(2): 210–215.

Hassibi, B. and Stork, D. G. (1993). Second order derivatives for network pruning: Optimal brain surgeon, *in* S. J. Hanson, J. Cowan and C. L. Giles (eds), *Advances in Neural Information Processing Systems 2*, Morgan Kaufmann, pp. 164–171.

Hastie, T. and Tibshirani, R. (2004). Efficient quadratic regularization for expression arrays, *Biostatistics* **5**(3): 329–340.

Hastie, T., Tibshirani, R. and Friedman, J. (2001). *The elements of statistical learning – Data mining, Inference, and Prediction*, Springer.

Haykin, S. (1999). *Neural networks – A comprehensive foundation*, Prentice Hall, Inc.

Hinton, G. and Sejnowski, T. J. (eds) (1999). *Unsupervised learning – Foundations of neural computation*, The MIT Press.

Hoegaerts, L., Suykens, J. A. K., Vandewalle, J. and De Moor, B. (2004). A comparison of pruning algorithms for sparse least squares support vector machines, *in* N. R. Pal, N. Kasabov, R. K. Mudi, S. Pal and S. K. Parui (eds), *Proceedings of the International Conference on Neural Information Processing (ICONIP 2004)*, Vol. 3316 of Lecture Notes in Computer Science, Springer-Verlag, pp. 1247–1253.

Hoerl, A. E. and Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems, *Technometrics* **12**(1): 55–67.

Hollmén, J. and Tikka, J. (2007). Compact and understandable descriptions of mixtures of bernoulli distributions, *in* M. R. Berthold, J. Shawe-Taylor and N. Lavrač (eds), *Proceedings of the 7th International Symposium on Intelligent Data Analysis (IDA 2007)*, Vol. 4723 of Lecture Notes in Computer Science, Springer-Verlag, pp. 1–12.

Hornik, K., Stinchcombe, M. and White, H. (1989). Multilayer feedforward networks are universal approximators, *Neural Networks* **2**(5): 359–366.

Hornik, K., Stinchcombe, M. and White, H. (1990). Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks, *Neural Networks* **3**(5): 551–560.

Hsieh, W. W. (2007). Nonlinear principal component analysis of noisy data, *Neural Networks* **20**(4): 434–443.

Huang, G.-B., Saratchandran, P. and Sundararajan, N. (2005). A generalized growing and pruning RBF (GGAP-RBF) neural network for function approximation, *IEEE Transactions on Neural Networks* **16**(1): 57–67.

Hyvärinen, A. (1999). Fast and robust fixed-point algorithms for independent component analysis, *IEEE Transactions on Neural Networks* **10**(3): 626–634.

Hyvärinen, A., Karhunen, J. and Oja, E. (2001). *Independent component analysis*, John Wiley & Sons, Inc.

Hyvärinen, A. and Oja, E. (2000). Independent component analysis: Algorithms and applications, *Neural Networks* **13**(4–5): 411–430.

Hyvärinen, A. and Pajunen, P. (1999). Nonlinear independent component analysis: Existence and uniqueness results, *Neural Networks* **12**(3): 429–439.

Ichino, M. and Yaguchi, H. (1994). Generalized Minkowski metrics for mixed feature-type data analysis, *IEEE Transactions on Systems, Man, and Cybernetics* **24**(4): 698–708.

Jain, A. K., Murty, M. N. and Flynn, P. J. (1999). Data clustering: A review, *ACM Computing Surveys* **31**(3): 264–323.

Japkowicz, N., Hanson, S. J. and Gluck, M. A. (2000). Nonlinear autoassociation is not equivalent to PCA, *Neural Computation* **12**(3): 531–545.

John, G. H., Kohavi, R. and Pfleger, K. (1994). Irrelevant features and the subset selection problem, *in* W. W. Cohen and H. Hirsh (eds), *Proceedings of the 11th International Conference on Machine Learning (ICML 1994)*, pp. 121–129.

Jutten, C. and Karhunen, J. (2004). Advances in blind source separation (BSS) and independent component ananlysis (ICA) for nonlinear mixtures, *International Journal of Neural Systems* **14**(5): 267–292.

Kiiveri, H. T. (2008). A general approach to simultaneous model fitting and variable elimination in response models for biological data with many more variables than observations, *BMC Bioinformatics* **9**(195).

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection, *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI 1995)*, Vol. 2, pp. 1137–1143.

Kohavi, R. and John, G. H. (1997). Wrappers for feature subset selection, *Artificial Intelligence* **97**(1-2): 273–324.

Kolman, E. and Margaliot, M. (2005). Are neural networks white boxes?, *IEEE Transactions on Neural Networks* **16**(5): 844–852.

Kramer, M. A. (1991). Nonlinear principal component analysis using autoassociative neural networks, *AIChE Jornal* **37**(2): 233–243.

Larsen, J. and Hansen, L. K. (1994). Generalization performance of regularized neural network models, *Proceedings of the IEEE International Workshop on Neural Networks for Signal Processing (NNSP 1994)*, pp. 42–51.

Law, M. H. C., Figueiredo, M. A. T. and Jain, A. K. (2004). Simultaneous feature selection and clustering using mixture models, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **26**(9): 1154–1166.

Le Cun, Y., Denker, J. S. and Solla, S. A. (1990). Optimal brain damage, *in* D. S. Touretzky (ed.), *Advances in Neural Information Processing Systems 2*, Morgan Kaufmann, pp. 598–605.

Leray, P. and Gallinari, P. (1999). Feature selection with neural networks, *Behaviormetrika* **26**(1): 145–166.

Li, K. and Peng, J.-X. (2007). Neural input selection–A fast model-based approach, *Neurocomputing* **70**.

Lin, B. Y. and Zhang, H. H. (2006). Component selection and smoothing in multivariate nonparametric regression, *The Annals of Statistics* **34**(5): 2272–2297.

Lowe, D. and Tipping, M. (1996). Feed-forward neural networks and topographic mappings for exploratory data analysis, *Neural Computing & Applications* **4**(2): 83–95.

MacKay, R. J. (1977). Variable selection in multivariate regression: An application of simultaneous test procedures, *Journal of the Royal Statistical Society. Series B (Methodological)* **39**(3): 371–380.

MacKay, R. J. and Oldford, R. W. (2000). Scientific method, statistical method and the speed of light, *Statistical Science* **15**(3): 254–278.

Malioutov, D., Çetin, M. and Willsky, A. S. (2005). A sparse signal reconstruction perspective for source localization with sensor arrays, *IEEE Transactions on Signal Processing* **53**(8): 3010–3022.

Mallows, C. L. (2000). Some comments on C_p, *Technometrics* **42**(1): 87–94. Reproduced with permission of the copyright owner. American Statistical Association and the American Society for Quality. ©1973.

Mannila, H. (1997). Methods and problems in data mining, *in* F. N. Afrati and P. G. Kolaitis (eds), *Proceedings of the 6th International Conference on Database Theory (ICDT 1997)*, Vol. 1186 of Lecture Notes in Computer Science, Springer-Verlag, pp. 41–55.

Matsubara, Y., Kikuchi, S., Sugimoto, M. and Tomita, M. (2006). Parameter estimation for stiff equations of biosystems using radial basis function networks, *BMC Bioinformatics* **7**: 230.

Maulik, U. and Bandyopadhyay, S. (2002). Performance evalution of some clustering algorithms and validity indices, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24**(12): 1650–1654.

McLachlan, G. and Peel, D. (2000). *Finite mixture models*, John Wiley & Sons.

Mika, S., Rätsch, G., Weston, J., Schölkopf, B. and Müller, K.-R. (1999). Fisher discriminant analysis with kernels, *Proceedings of the IEEE International Workshop on Neural Networks for Signal Processing (NNSP 1999)*, pp. 41–48.

Miller, A. J. (1990). *Subset selection in regression*, Chapman and Hall.

Milton, J. S. and Arnold, J. C. (1990). *Introduction to probability and statistics*, McGraw-Hill.

Møller, M. F. (1993). A scaled conjugate gradient algorithm for fast supervised learning, *Neural Networks* **6**(4): 525–533.

Moody, J. E. (1991). Note on generalization, regularization, and architecture selection in nonlinear learning systems, *Proceedings of the IEEE International Workshop on Neural Networks for Signal Processing (NNSP 1991)*, pp. 1–10.

Moody, J. E. (1992). The effective number of parameters: An analysis of generalization and regularization in nonlinear learning systems, *in* J. E. Moody, S. J. Hanson and R. P. Lippmann (eds), *Advances in Information Processing Systems 4*, Morgan Kaufmann, pp. 847–854.

Mooney, R. J. and Bunescu, R. (2007). Mining knowledge from text using information extraction, *ACM SIGKDD Explorations Newsletter* **7**(1): 3–10.

Myers, R. H. and Montgomery, D. C. (1997). A tutorial on generalized linear models, *Journal of Quality Technology* **29**(3): 274–291.

Myllykangas, S., Tikka, J., Böhling, T., Knuutila, S. and Hollmén, J. (2008). Classification of human cancers based on DNA copy number amplification modeling, *BMC Medical Genomics* **1**(15).

Nair, P. B., Choudhury, A. and Keane, A. J. (2002). Sparse greedy learning algorithms for sparse regression and classification with Mercer kernels, *Journal of machine learning research* **3**: 781–801.

Narendra, P. M. and Fugunaga, K. (1977). A branch and bound algorithm for feature selection, *IEEE Transactions on Computers* **C-26**(9): 917–922.

Oja, E. (1991). Data compression, feature extraction, and autoassociation in feedforward neural networks, *in* T. Kohonen, K. Mäkisara, O. Simula and J. Kangas (eds), *Proceedings of the 1st International Conference on Neural Networks (ICANN 1991)*, Artificial Neural Networks, Elsevier Science Publishers, pp. 737–745.

Oja, E. (2002). Unsupervised learning in neural computation, *Theoretical Computer Science* **287**: 187–207.

Orr, M. J. (1996). Introduction to radial basis functions networks, *Technical report*, Edinburgh University, Edinburgh, Scotland, UK.

Orr, M. J. L. (1995). Regularization in the selection of radial basis function centers, *Neural Computation* **7**(3): 606–623.

Osborne, M. R., Presnell, B. and Turlach, B. A. (2000). A new approach to variable selection in least squares problems, *IMA Journal of Numerical Analysis* **20**(3): 389–404.

Park, J. and Sandberg, I. W. (1991). Universal approximation using radial-basis-function networks, *Neural Computation* **3**(2): 246–257.

Park, J. and Sandberg, I. W. (1993). Approximation and radial-basis-function networks, *Neural Computation* **5**(2): 305–316.

Parzen, E. (1962). On estimation of a probability density function and mode, *The Annals of Mathematical Statistics* **33**(3): 1065–1076.

Peng, J.-X., Li, K. and Irwin, G. W. (2007). A novel continuous forward algorithm for RBF neural modelling, *IEEE Transactions on Automatic Control* **52**(1): 117–122.

Pirooznia, M., Yang, J. Y., Yang, M. Q. and Deng, Y. (2008). A comparative study of different machine learning methods on microarray gene expression data, *BMC Genomics* **9**(Suppl 1): S13.

Poggio, T. and Girosi, F. (1990). Networks for approximation and learning, *Proceedings of the IEEE* **78**(9): 1481–1497.

Rakotomamonjy, A. (2003). Variable selection using SVM-based criteria, *Journal of Machine Learning Research* **3**: 1357–1370.

Rakotomamonjy, A. (2007). Analysis of SVM regression bounds for variable ranking, *Neurocomputing* **70**(7–9): 1489–1501.

Redner, R. A. and Walker, H. F. (1984). Mixture densities, maximum likelihood and the EM algorithm, *SIAM Review* **26**(2): 195–239.

Reed, R. (1993). Pruning algorithms – A survey, *IEEE Transactions on Neural Networks* **4**(5): 740–747.

Refenes, A.-P. N. and Zapranis, A. D. (1999). Neural model identification, variable selection and model adequacy, *Journal of Forecasting* **18**(5): 299–332.

Reunanen, J. (2003). Overfitting in making comparisons between variable selection methods, *Journal of Machine Learning Reasearch* **3**: 1371–1382.

Rivals, I. and Personnaz, L. (2003). MLPs (mono-layer polynomials and multilayer perceptrons) for nonlinear modeling, *Journal of Machine Learning Research* **3**: 1383–1398.

Roberts, K. (2008). The neural network analysis of international organizational performance, *Economics, Management, and Financial Markets* **3**(1): 11–23.

Ruck, D. W., Rogers, S. K. and Kabrisky, M. (1990). Feature selection using a multilayer perceptron, *Journal of Neural Network Computing* **2**(2): 40–48.

Rumelhart, D. E., Hinton, G. E. and Williams, R. J. (1986). Learning representations by back-propagating errors, *Nature* **323**: 533–536.

Sammon, J. W. (1969). A nonlinear mapping for data structure analysis, *IEEE Transactions on Computers* **C-18**(5): 401–409.

Saunders, C., Gammerman, A. and Vovk, V. (1998). Ridge regression learning algorithm in dual variables, *in* J. W. Shavlik (ed.), *Proceedings of the Fifteenth International Conference on Machine Learning (ICML 1998)*, pp. 515–521.

Schmidt, M., Fung, G. and Rosales, R. (2007). Fast optimization methods for L1 regularization: A comparative study and two new approaches, *in* J. N. Kok, J. Koronacki, R. Lopez de Mantaras, S. Matwin, D. Mladenič and A. Skowron (eds), *Proceedings of the 18th European Conference on Machine Learning (ECML 2007)*, Vol. 4701 of Lecture Notes in Computer Science, Springer-Verlag, pp. 286–297.

Schölkopf, B., Smola, A. and Müller, K.-R. (1998). Nonlinear component analysis as a kernel eigenvalue problem, *Neural Computation* **10**(5): 1299–1319.

Schumacher, M., Roßner, R. and Vach, W. (1996). Neural networks and logistic regression: Part I, *Computational Statistics & Data Analysis* **21**(6): 661–682.

Shao, J. (1993). Linear model selection by cross-validation, *Journal of the American Statistical Association* **88**(422): 486–494.

Shashua, A. (1999). On the relantionship between the support vector machine for classification and sparsified Fisher's linear discriminant, *Neural Processing Letters* **9**(2): 129–139.

Similä, T. (2007a). *Advances in variable selection and visualization methods for analysis of multivariate data*, PhD thesis, Helsinki University of Technology.

Similä, T. (2007b). Majorize-minimize algorithm for multiresponse sparse regression, *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2007)*, pp. 553–556.

Similä, T. and Tikka, J. (2008). Combined input variable selection and model complexity control for nonlinear regression, *Pattern Recognition Letters* . Accepted for publication.

Smola, A. J. and Schölkopf, B. (2000). Sparse greedy matrix approximation for machine learning, *in* P. Langley (ed.), *Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000)*, pp. 911–918.

Smola, A. J. and Schölkopf, B. (2004). A tutorial on support vector regression, *Statistics and Computing* **14**(3): 199–222.

Somol, P., Pudil, P. and Kittler, J. (2004). Fast branch & bound algorithms for optimal feature selection, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **26**(7): 900–912.

Somol, P., Pudil, P., Novovičová, J. and Palík, P. (1999). Adaptive floating search methods in feature selection, *Pattern Recognition Letters* **20**(11–13): 1157–1163.

Sparks, R. S., Zucchini, W. and Courtsourides, D. (1985). On variable selection in multivariate regression, *Communications in Statistics. Theory and Methods* **14**(7): 1569–1578.

Specht, D. F. (1991). A general regression neural network, *IEEE Transactions on Neural Networks* **2**(6): 568–576.

Suykens, J. A. K., De Brabanter, J., Lukas, L. and Vandewalle, J. (2002). Weighted least squares support vector machines: robustness and sparse approximation, *Neurocomputing* **48**(1–4): 85–105.

Suykens, J. A. K., Van Gestel, T., De Brabanter, J., De Moor, B. and Vandewalle, J. (2002). *Least squares support vector machines*, World Scientific Publishing.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso, *Journal of the Royal Statistical Society, Series B (Methodological)* **58**(1): 267–288.

Tikka, J., Hollmén, J. and Myllykangas, S. (2007). Mixture modeling of DNA copy number amplification patterns in cancer, *in* F. Sandoval, A. Prieto, J. Cabestany and M. Graña (eds), *Proceedings of the 9th International Work-Conference on Artificial Neural Networks (IWANN 2007)*, Vol. 4507 of Lecture Notes in Computer Science, Springer-Verlag, pp. 972–979.

Tipping, M. E. (2001). Sparse kernel principal component analysis, *in* T. K. Leen, T. G. Dietterich and V. Tresp (eds), *Advances in Neural Information Processing Systems 13*, The MIT Press, pp. 633–639.

Tipping, M. E. and Lowe, D. (1998). Shadow targets: A novel algorithm for topographic projections by radial basis functions, *Neurocomputing* **19**(1–3): 211–222.

Tomenko, V., Ahmed, S. and Popov, V. (2007). Modelling constructed wetland treatment system performance, *Ecological Modelling* **205**(3–4): 355–364.

Tropp, J. A. (2006). Algorithms for simultaneous sparse approximation. Part II: Convex relaxation, *Signal Processing* **86**(3): 589–602.

Tukey, J. W. (1977). *Exploratory data analysis*, Addison-Wesley.

Tukey, J. W. (1980). We need both exploratory and confirmatory, *The American Statistician* **34**(1): 23–25.

Turlach, B. A. (2004). Discussion of least angle regression, *The Annals of Statistics* **32**(2): 481–490.

Turlach, B. A., Williams, W. N. and Wright, S. J. (2005). Simultaneous variable selection, *Technometrics* **47**(3): 349–363.

Vach, W., Roßner, R. and Schumacher, M. (1996). Neural networks and logistic regression: Part II, *Computational Statistics & Data Analysis* **21**(6): 683–701.

Van Gestel, T., Suykens, J. A. K., De Moor, B. and Vandewalle, J. (2001). Automatic relevance determination for least squares support vector machine regression, *Proceedings of the International Joint Conference on Neural Networks (IJCNN 2001)*, Vol. 4, pp. 2416–2421.

Verleysen, M. (2003). Learning high-dimensional data, *in* S. Ablameyko, L. Goras, M. Gori and V. Piuri (eds), *Limitations and Future Trends in Neural Computation*, IOS Press, pp. 141–162.

Villmann, T., Schleif, F. and Hammer, B. (2006). Comparison of relevance learning vector quantization with other metric adaptive classification methods, *Neural Networks* **19**(5): 610–622.

Webb, A. R. (1995). Multidimensional scaling by iterative majorization using radial basis functions, *Pattern Recognition* **28**(5): 753–759.

Wei, W. W. S. (2006). *Time series analysis – Univariate and multivariate methods*, Pearson Education, Inc.

Weston, J., Mukherjee, S., Chapelle, O., Pontil, M., Poggio, T. and Vapnik, V. (2001). Feature selection for SVMs, *in* T. K. Leen, T. G. Dietterich and V. Tresp (eds), *Advances in Neural Information Processing Systems 13*, The MIT Press.

White, H. (1990). Connectionist nonparametric regression: Multilayer feedforward networks can learn arbitrary mappings, *Neural Networks* **3**(5): 535–549.

Xu, R. and Wunsch, D. (2005). Survey of clustering algorithms, *IEEE Transactions on Neural Networks* **16**(3): 645–678.

Yacoub, M. and Bennani, Y. (2000). Features selection and architecture optimization in connectionist systems, *International Jornal of Neural Systems* **10**(5): 379–395.

Yuan, M. and Lin, Y. (2006). Model selection and estimation in regression with grouped variables, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **68**(1): 49–67.

Yuan, M. and Lin, Y. (2007). On the nonnegative garrotte estimator, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **69**(2): 143–161.

Zhang, G. P. (2007). Avoiding pitfalls in neural network research, *IEEE Transactions on Systems, Man, and Cybernetics – Part C: Applications and Reviews* **37**(1): 3–16.

Zhu, J. and Hastie, T. (2005). Kernel logistic regression and the import vector machine, *Journal of Computational and Graphical Statistics* **14**(1): 185–205.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **67**(2): 301–320.