



# Least Squares Support Vector Regression with Applications to Large-Scale Data: a Statistical Approach

**Kris De Brabanter**

Promotor:  
Prof. dr. ir. B. De Moor

Co-Promotor:  
Prof. dr. ir. J. Suykens

Proefschrift voorgedragen tot  
het behalen van het doctoraat  
in de ingenieurswetenschappen





**KATHOLIEKE UNIVERSITEIT LEUVEN**  
Faculteit Ingenieurwetenschappen  
Departement Elektrotechniek  
Afdeling SISTA - SCD  
Kasteelpark Arenberg 10, B-3001 Leuven, Belgium

# **Least Squares Support Vector Regression with Applications to Large-Scale Data: a Statistical Approach**

**Kris De Brabanter**

Jury:

Prof. dr. ir. Y. Willems, chairman  
(Departement Computerwetenschappen)  
Prof. dr. ir. B. De Moor, promotor  
(Departement Elektrotechniek)  
Prof. dr. ir. J. Suykens, co-promotor  
(Departement Elektrotechniek)  
Prof. dr. ir. J. Vandewalle  
(Departement Elektrotechniek)  
Prof. dr. ir. J. Van Impe  
(Departement Chemische Ingenieurstechnieken)  
Prof. dr. I. Gijbels  
(Departement Wiskunde, LStat)  
Prof. dr. N. Veraverbeke  
(Center for Statistics, Hasselt University)  
Prof. dr. L. Györfi  
(Budapest University)

Proefschrift voorgedragen tot  
het behalen van het doctoraat  
in de ingenieurwetenschappen

April 2011

© Katholieke Universiteit Leuven – Faculty of Engineering  
Kasteelpark Arenberg 10, B-3001 Leuven (Belgium)

Alle rechten voorbehouden. Niets uit deze uitgave mag worden vermenigvuldigd en/of openbaar gemaakt worden door middel van druk, fotocopie, microfilm, elektronisch of op welke andere wijze ook zonder voorafgaande schriftelijke toestemming van de uitgever.

All rights reserved. No part of the publication may be reproduced in any form by print, photoprint, microfilm or any other means without written permission from the publisher.

D/2011/7515/53  
ISBN 978-94-6018-351-5

# Foreword

Beyond the scientific aspects of this work, I would like to highlight how important it has been to be supported by my colleagues, family and friends.

I start by expressing my gratitude to my promotor Prof. dr. ir. Bart De Moor for giving me the opportunity to do this PhD work. I am thankful for the freedom and great support he gave me during my PhD. I would also like to acknowledge my co-promotor Prof. dr. ir. Johan Suykens for proposing numerous interesting research topics. Also, the many technical discussions were particularly useful and inspiring. Both promotors were a huge support and a source of inspiration for my research.

I would like to thank the members of the jury: Prof. (em.) dr. ir. Yves Willems, Prof. dr. ir. Joos Vandewalle, Prof. dr. ir. Jan Van Impe, Prof. dr. Irène Gijbels, Prof. dr. Noël Veraverbeke and Prof. dr. László Györfi. I thank them for their guidance, new insights and the revision of my manuscript. I thank Prof. dr. ir. Jan Van Impe for many useful discussions and for his support during my PhD. I am also very thankful to Prof. dr. Irène Gijbels for answering my endless questions regarding nonparametric statistics and teaching me new insights in this field. Further, I would like to thank Prof. dr. László Györfi with whom I had the pleasure of working with. During this time he taught me a lot about nonparametric regression, density and entropy estimation. Finally, I am grateful to Prof. dr. Noël Veraverbeke to be part of the jury and Prof. (em.) dr. ir. Yves Willems for being the chairman of this examination committee.

I would like to thank my colleagues and friends in my research group for the pleasant environment: Philippe, Peter, Tony, Raf, Kim, Pieter, Kristiaan and many other wonderful people that I never forget. I definitely want to acknowledge Lut and Ida without whose help I would not have succeeded in organizing many steps of my PhD. Also a special thanks goes to John who took care of all my financial arrangements accompanying my PhD and to Maarten and Liesbeth who are responsible for the IT and webdesign within SISTA. They were always prepared to solve practical questions and problems.

My gratitude also goes to my parents and grandparents for their continuing support during my PhD years. Finally, I acknowledge my wife Sahar who supported me from the very first moment of my PhD till the final defense. Thank you!

Kris De Brabanter  
Heverlee, April 2011

# Beknopte Samenvatting

Niet-parametrische regressie is een krachtige methode voor data analyse omdat deze techniek weinig assumpties oplegt aan de vorm van de gemiddelde functie. Deze technieken zijn uiterst geschikt voor het ontdekken van niet lineaire verbanden tussen variabelen. Een nadeel van deze methodes is hun rekencomplexiteit wanneer grote data sets worden beschouwd. Reductie van de complexiteit voor kleinste kwadraten support vector machines (LS-SVM) is mogelijk door gebruik te maken van vaste-grootte kleinste kwadraten support vector machines (FS-LSSVM). Deze methode is geschikt voor behandelen van grote data sets op een PC.

De eigenschappen van LS-SVM worden bestudeerd wanneer de Gauss-Markov condities niet vervuld zijn. We ontwikkelen een robuuste versie van LS-SVM gebaseerd op iteratieve herweging waarbij de gewichten gebaseerd zijn op de distributie van de residuen. We tonen aan dat de empirische maxbias curve van de ontwikkelde robuuste procedure slechts licht verhoogt als functie van het aantal uitbijters. Verder, stellen we drie conditions voor om een volledig robuuste procedure te verkrijgen.

Verder worden de gevolgen bestudeerd wanneer de onafhankelijk en identisch verdeelde assumptie niet vervuld is. We tonen voor niet-parametrische kernel gebaseerde regressie aan dat de methodes voor model selectie falen wanneer de data gecorreleerd is. We ontwikkelen een model selectie procedure voor LS-SVM die bestand is tegen correlatie. Hierbij is geen enkele voorkennis van de correlatiestructuur vereist.

Vervolgens, ontwikkelen we bias gecorrigeerde  $100(1 - \alpha)\%$  benaderende betrouwbaarheids- en predictie-intervallen voor lineaire smoothers (regressie en classificatie). We tonen, onder bepaalde condities, de asymptotische normaliteit van LS-SVM aan. Verder, worden d.m.v. voorbeelden het praktische nut van deze intervallschattingen geïllustreerd voor regressie en classificatie.

Tot slot, worden er een aantal toepassingen gegeven i.v.m. systeemidentificatie, hypothesetesten en dichtheidsfunctieschatting gebaseerd op de ontwikkelde technieken.





# Abstract

Nonparametric regression is a very popular tool for data analysis because these techniques impose few assumptions about the shape of the mean function. Hence, they are extremely flexible tools for uncovering nonlinear relationships between variables. A disadvantage of these methods is their computational complexity when considering large data sets. In order to reduce the complexity for least squares support vector machines (LS-SVM), we propose a method called Fixed-Size LS-SVM which is capable of handling large data set on standard personal computers.

We study the properties of the LS-SVM regression when relaxing the Gauss-Markov conditions. We propose a robust version of LS-SVM based on iterative reweighting with weights based on the distribution of the error variables. We show that the empirical maxbias of the proposed robust estimator increases slightly with the number of outliers in region and stays bounded right up to the breakdown point. We also establish three conditions to obtain a fully robust nonparametric estimator.

We investigate the consequences when the i.i.d. assumptions is violated. We show that, for nonparametric kernel based regression, classical model selection procedures such as cross-validation, generalized cross-validation and  $v$ -fold cross-validation break down in the presence of correlated data and not the chosen smoothing method. Therefore, we develop a model selection procedure for LS-SVM in order to effectively handle correlation in the data without requiring any prior knowledge about the correlation structure.

Next, we propose bias-corrected  $100(1 - \alpha)\%$  approximate confidence and prediction intervals (pointwise and uniform) for linear smoothers, in particularly for LS-SVM. We prove, under certain conditions, the asymptotic normality of LS-SVM. Further, we show the practical use of these interval estimates by means of toy examples for regression and classification.

Finally, we illustrate the capabilities of the proposed methods on a number of applications i.e. system identification, hypothesis testing and density estimation.



# List of Abbreviations

ARX	AutoRegressive with eXogenous inputs
MSE	Mean Squared Error
RMSE	Root Mean Squared Error
i.i.d.	Independent and Identically Distributed
NW	Nadaraya-Watson
SVM	Support Vector Machine
QP	Quadratic Programming
LS-SVM	Least Squares Support Vector Machine
RSS	Residual Sum of Squares
MLP	Multi-Layer Perceptron
CV	Cross-Validation
LOO-CV	Leave-One-Out Cross-Validation
GCV	Generalized Cross-Validation
ISE	Integrated Squared Error
FPE	Final Prediction Error
AIC	Akaike Information Criterion
BIC	Bayesian Information Criterion
HQIC	Hannan-Quin Information Criterion
TIC	Takeuchi's Information Criterion
MDL	Minimum Description Length
KL	Kullback-Leibler
AICC	Akaike Information Criterion Corrected
CSA	Coupled Simulated Annealing
CLM	Coupled Local Minimizers
SA	Simulated Annealing
FS-LSSVM	Fixed-Size Least Squares Support Vector Machines
PV	Prototype Vectors
MISE	Mean Integrated Squared Error

AMISE	Asymptotic Mean Integrated Squared Error
IFGT	Improved Fast Gauss Transform
SVR	Support Vector Regression
SVC	Support Vector Classification
Q-Q	Quantile-Quantile
IF	Influence Function
BP	Breakdown Point
SiZer	Significant ZERO crossings of derivatives
MAD	Median Absolute Deviation
LS	Least Squares
KBR	Kernel Based Regression
RKHS	Reproducing Kernel Hilbert Space
TV	Total Variation
IRLS-SVM	Iteratively Reweighted LS-SVM
ML	Maximum Likelihood
MASE	Mean Asymptotic Squared Error
AR	Auto Regressive
MCV	Modified or leave- $(2l + 1)$ -out Cross-Validation
CC-CV	Correlation-Corrected Cross-Validation
KKT	Karush-Kuhn-Tucker
CI	Confidence Interval
PI	Prediction Interval

# List of Symbols

$a \in A$	$a$ is an element of the set $A$
$A \subseteq B$	Set $A$ is contained in the set $B$
$A \subset B$	$A \subseteq B$ and $A \neq B$
$m(x) = \mathbf{E}[Y X = x]$	Regression function
$\hat{m}_n$	Regression estimate
$\mathbb{N}$	Set of all natural numbers
$\mathbb{N}_0$	Set of nonnegative integers
$\mathbb{R}$	Set of real numbers
$\mathbb{R}^d$	Set of $d$ -dimensional real numbers
$K : \mathbb{R}^d \rightarrow \mathbb{R}$	Kernel function
$h > 0$	Bandwidth of the kernel
$f : C \rightarrow D$	A function $f$ from $C$ to $D$
$\mathbf{E}[X]$	Expected value of $X$
$C(\mathbb{R}^d)$	Set of all continuous functions $f : \mathbb{R}^d \rightarrow \mathbb{R}$
$\mathcal{D}_n = \{(X_1, Y_1), \dots, (X_n, Y_n)\}$	Training data
$\mathcal{R}$	Risk functional
$\mathcal{R}_{\text{emp}}$	Empirical risk functional
$\mathbf{Cov}[X, Y]$	Covariance of $X$ and $Y$
$\mathbf{Var}[X]$	Variance of $X$
$x^{(1)}, \dots, x^{(d)}$	Components of the $d$ -dimensional column vector $x$
$\ x\  = \sqrt{\sum_{i=1}^d (x^{(i)})^2}$	Euclidean norm of $x \in \mathbb{R}^d$
$\ f\  = \sqrt{\int f(x)^2 dF(x)}$	$L_2$ norm of $f : \mathbb{R}^d \rightarrow \mathbb{R}$
$\ f\ _\infty = \sup_{x \in \mathbb{R}^d}  f(x) $	Supremum norm of $f : \mathbb{R}^d \rightarrow \mathbb{R}$
$\mathbf{P}[A]$	Probability of an event $A$
$\Gamma(x)$	Gamma function evaluated in point $x$
$u = \arg \min_{x \in D} f(x)$	Abbreviation for $u \in D$ and $f(u) = \min_{x \in D} f(x)$

$\mathcal{N}(\mu, \sigma^2)$	Normal distribution with mean $\mu$ and variance $\sigma^2$
$\mathcal{U}[a, b]$	Uniformly distributed over the interval $[a, b]$
$C^v(\mathbb{R})$	Set of all $v$ -times differentiable functions
$L^2$	Space of square integrable functions
$\gamma$	Regularization constant of LS-SVM
$\text{tr}[A]$	Trace of the matrix $A$
$ A $	Determinant of the matrix $A$
$O, o$	Order of magnitude symbols
$\sim$	Asymptotically equal
$I_A(x) = I_{x \in A}$	Indicator function of a set $A$
$\varphi$	Nonlinear mapping from input to feature space
$\xrightarrow{d}$	Convergence in distribution
$\xrightarrow{P}$	Convergence in probability
$f^{(r)}$	$r$ th derivative of $f$
$\mathcal{G}_\epsilon$	Gross-error model with $0 \leq \epsilon \leq 1$
$T(F)$	Statistical functional
$T(\hat{F}_n)$	Statistical functional of the sample distribution
$\tilde{K} : \mathbb{R}^d \rightarrow \mathbb{R}$	Bimodal kernel function satisfying $K(0) = 0$
$\tilde{K}_\epsilon : \mathbb{R}^d \rightarrow \mathbb{R}$	$\epsilon$ -optimal class of bimodal kernels
$[z]_+$	$\max(z, 0)$
$\mathbf{1}_n$	Vector of ones

# Publication list

## Journal papers

- [1] Falck, T., Dreesen, P., **De Brabanter, K.**, Pelckmans, K., De Moor, B., Suykens, J.A.K., Least-Squares Support Vector Machines for the Identification of Wiener-Hammerstein Systems, *Submitted*, 2011.
- [2] **De Brabanter K.**, De Brabanter J., Suykens J.A.K., De Moor B., Kernel Regression in the Presence of Correlated Errors, *Submitted*, 2011.
- [3] **De Brabanter K.**, Karsmakers P., De Brabanter J., Suykens J.A.K., De Moor B., Confidence Bands for Least Squares Support Vector Machine Classifiers: A Regression Approach, *Submitted*, 2010.
- [4] **De Brabanter K.**, De Brabanter J., Suykens J.A.K., De Moor B., Approximate Confidence and Prediction Intervals for Least Squares Support Vector Regression, *IEEE Transactions on Neural Networks*, 22(1):110–120 , 2011.
- [5] Sahnaf S., **De Brabanter K.**, Degraeve R., Suykens J.A.K., De Moor B., Groeseneken G., Modelling of Charge Trapping/De-trapping Induced Voltage Instability in High-k Gate Dielectrics, *Submitted*, 2010.
- [6] Karsmakers P., Pelckmans K., **De Brabanter K.**, Van Hamme H., Suykens J.A.K., Sparse Conjugate Directions Pursuit with Application to Fixed-size Kernel Models, *Submitted*, 2010.
- [7] Sahnaf S., Degraeve R., Cho M., **De Brabanter K.**, Roussel Ph.J., Zahid M.B., Groeseneken G., Detailed Analysis of Charge Pumping and  $I_d - V_g$  Hysteresis for Profiling Traps in  $\text{SiO}_2/\text{HfSiO}(\text{N})$ , *Microelectronic Engineering*, 87(12):2614–2619, 2010.
- [8] **De Brabanter K.**, De Brabanter J., Suykens J.A.K., De Moor B., Optimized Fixed-Size Kernel Models for Large Data Sets, *Computational Statistics & Data Analysis*, 54(6):1484–1504, 2010.

## Conference papers

- [1] López J., **De Brabanter K.**, Dorransoro J.R., Suykens J.A.K., Sparse LS-SVMs with  $L_0$ -Norm Minimization, *Accepted for publication in Proc. of the 19th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, Brugge (Belgium), 2011.
- [2] Huyck B., **De Brabanter K.**, Logist F., De Brabanter J., Van Impe J., De Moor B., Identification of a Pilot Scale Distillation Column: A Kernel Based Approach, *Accepted for publication in 18th World Congress of the International Federation of Automatic Control (IFAC)*, 2011.
- [3] **De Brabanter K.**, Karsmakers P., De Brabanter J., Pelckmans K., Suykens J.A.K., De Moor B., On Robustness in Kernel Based Regression, *NIPS 2010 Robust Statistical Learning (ROBUSTML) (NIPS 2010)*, Whistler, Canada, December 2010.
- [4] **De Brabanter K.**, Sahhaf S., Karsmakers P., De Brabanter J., Suykens J.A.K., De Moor B., Nonparametric Comparison of Densities Based on Statistical Bootstrap, in *Proc. of the Fourth European Conference on the Use of Modern Information and Communication Technologies (ECUMICT)*, Gent, Belgium, March 2010, pp. 179–190.
- [5] **De Brabanter K.**, De Brabanter J., Suykens J.A.K., De Moor B., Kernel Regression with Correlated Errors, in *Proc. of the the 11th International Symposium on Computer Applications in Biotechnology (CAB)*, Leuven, Belgium, July 2010, pp. 13–18.
- [6] **De Brabanter K.**, Pelckmans K., De Brabanter J., Debruyne M., Suykens J.A.K., Hubert M., De Moor B., Robustness of Kernel Based Regression: a Comparison of Iterative Weighting Schemes, in *Proc. of the 19th International Conference on Artificial Neural Networks (ICANN)*, Limassol, Cyprus, September 2009, pp. 100–110.
- [7] **De Brabanter K.**, Dreesen P., Karsmakers P., Pelckmans K., De Brabanter J., Suykens J.A.K., De Moor B., Fixed-Size LS-SVM Applied to the Wiener-Hammerstein Benchmark, in *Proc. of the 15th IFAC Symposium on System Identification (SYSID 2009)*, Saint-Malo, France, July 2009, pp. 826–831.

## Software

- [1] **De Brabanter K.**, Karsmakers P., Ojeda F., Alzate C., De Brabanter J., Pelckmans K., De Moor B., Vandewalle J., Suykens J.A.K., LS-SVMlab Toolbox User’s Guide version 1.7”, Internal Report 10-146, ESAT-SISTA, K.U.Leuven (Leuven, Belgium), 2010.



# Contents

<b>Contents</b>	<b>xiii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Historical Evolution and General Background . . . . .	1
1.2 Practical Applications . . . . .	4
1.2.1 Biomedical Data . . . . .	4
1.2.2 Financial Data . . . . .	5
1.2.3 System Identification . . . . .	5
1.2.4 Time Series Analysis . . . . .	6
1.3 Organization and Contributions of the Thesis . . . . .	6
<b>2 Model Building</b>	<b>13</b>
2.1 Regression Analysis and Loss Functions . . . . .	13
2.2 Assumptions, Restrictions and Slow Rate . . . . .	16
2.3 Curse of Dimensionality . . . . .	18
2.4 Parametric and Nonparametric Regression Estimators: An Overview	21
2.4.1 Parametric Modeling . . . . .	21
2.4.2 Local Averaging . . . . .	22
2.4.3 Local Modeling . . . . .	25
2.4.4 Global Modeling . . . . .	27

2.4.5	Penalized Modeling . . . . .	27
2.5	Support Vector Machines . . . . .	28
2.5.1	Basic Idea of Support Vector Machines . . . . .	28
2.5.2	Primal-Dual Formulation of Support Vector Machines . . . . .	29
2.5.3	Least Squares Support Vector Machines . . . . .	32
2.6	Conclusions . . . . .	33
<b>3</b>	<b>Model Selection Methods</b>	<b>35</b>
3.1	Introduction . . . . .	35
3.2	Cross-Validation Procedures . . . . .	37
3.2.1	Cross-Validation Philosophy . . . . .	37
3.2.2	Leave-One-Out Cross-Validation . . . . .	38
3.2.3	$\mathbf{v}$ -fold Cross-Validation . . . . .	39
3.2.4	Generalized Cross-Validation . . . . .	40
3.3	Complexity Criteria: Final Prediction Error, AIC, Mallows' $\mathbf{C}_p$ and BIC . . . . .	41
3.4	Choosing the Learning Parameters . . . . .	43
3.4.1	General Remarks . . . . .	43
3.4.2	Optimization Strategy . . . . .	44
3.5	Conclusions . . . . .	46
<b>4</b>	<b>Fixed-Size LS-SVM</b>	<b>47</b>
4.1	Introduction . . . . .	47
4.2	Estimation in the Primal Space . . . . .	48
4.2.1	Finite Approximation to the Feature Map . . . . .	48
4.2.2	Solving the Problem in Primal Space . . . . .	50
4.3	Active Selection of a Subsample . . . . .	51
4.3.1	Subsample Based on Entropy Criteria . . . . .	51
4.3.2	Bandwidth Selection for Density Estimation . . . . .	53

- 4.3.3 Solve-the-Equation Plug-In Method . . . . . 56
- 4.3.4 Maximizing Rényi Entropy vs. Random Sampling . . . . . 57
- 4.4 Selecting the number of prototype vectors . . . . . 59
- 4.5 Fast  $\mathbf{v}$ -fold Cross-Validation for FS-LSSVM . . . . . 62
  - 4.5.1 Extended Feature Matrix Can Fit Into Memory . . . . . 62
  - 4.5.2 Extended Feature Matrix Cannot Fit Into Memory . . . . . 65
- 4.6 Computational Complexity and Numerical Experiments on  $v$ -fold CV 66
  - 4.6.1 Computational Complexity Analysis . . . . . 66
  - 4.6.2 Numerical Experiments . . . . . 67
- 4.7 Classification and Regression Results . . . . . 70
  - 4.7.1 Description of the Data Sets . . . . . 70
  - 4.7.2 Description of the Reference Algorithms . . . . . 71
  - 4.7.3 Performance of binary FS-LSSVM classifiers . . . . . 72
  - 4.7.4 Performance of multi-class FS-LSSVM classifiers . . . . . 74
  - 4.7.5 Performance of FS-LSSVM for Regression . . . . . 75
- 4.8 Conclusions . . . . . 76
- 5 Robustness in Kernel Based Regression 77**
  - 5.1 Introduction . . . . . 77
  - 5.2 Measures of Robustness . . . . . 79
    - 5.2.1 Influence Functions and Breakdown Points . . . . . 79
    - 5.2.2 Empirical Influence Functions . . . . . 82
  - 5.3 Residuals and Outliers in Regression . . . . . 86
    - 5.3.1 Linear Regression . . . . . 86
    - 5.3.2 Kernel Based Regression . . . . . 88
  - 5.4 Robustifying LS Kernel Based Regression . . . . . 89
    - 5.4.1 Problems with Outliers in Nonparametric Regression . . . . . 89
    - 5.4.2 Theoretical Background . . . . . 92

5.4.3	Application to Least Squares Support Vector Machines . . .	97
5.4.4	Weight Functions . . . . .	98
5.4.5	Speed of Convergence-Robustness Tradeoff . . . . .	102
5.4.6	Robust Selection of Tuning Parameters . . . . .	102
5.5	Simulations . . . . .	104
5.5.1	Empirical Maxbias Curve . . . . .	104
5.5.2	Toy example . . . . .	105
5.5.3	Real Life Data Sets . . . . .	106
5.6	Conclusions . . . . .	107
<b>6</b>	<b>Kernel Regression with Correlated Errors</b>	<b>109</b>
6.1	Introduction . . . . .	109
6.2	Problems with Correlation . . . . .	111
6.3	New Developments in Kernel Regression with Correlated Errors . .	112
6.3.1	No Positive Definite Kernel Constraint . . . . .	113
6.3.2	Positive Definite Kernel Constraint . . . . .	119
6.3.3	Drawback of Using Bimodal Kernels . . . . .	121
6.4	Simulations . . . . .	125
6.4.1	CC-CV vs. LOO-CV with Different Noise Models . . . . .	125
6.4.2	Evolution of the Bandwidth Under Correlation . . . . .	127
6.4.3	Comparison of Different Bimodal Kernels . . . . .	128
6.4.4	Real life data sets . . . . .	130
6.5	Conclusions . . . . .	131
<b>7</b>	<b>Confidence and Prediction Intervals</b>	<b>133</b>
7.1	Introduction . . . . .	133
7.2	Estimation of Bias and Variance . . . . .	136
7.2.1	LS-SVM Regression and Smoother Matrix . . . . .	136

7.2.2	Bias Estimation . . . . .	138
7.2.3	Variance Estimation . . . . .	141
7.3	Confidence and Prediction Intervals: Regression . . . . .	144
7.3.1	Pointwise Confidence Intervals . . . . .	144
7.3.2	Simultaneous Confidence Intervals . . . . .	147
7.3.3	Pointwise and Simultaneous Prediction Intervals . . . . .	153
7.4	Bootstrap Based Confidence and Prediction Intervals . . . . .	153
7.4.1	Bootstrap Based on Residuals . . . . .	154
7.4.2	Construction of Bootstrap Confidence and Prediction Intervals	154
7.5	Simulations: The Regression Case . . . . .	156
7.5.1	Empirical Coverage Probability . . . . .	156
7.5.2	Homoscedastic Examples . . . . .	157
7.5.3	Heteroscedastic Examples and Error Variance Estimation .	159
7.6	Confidence Intervals: Classification . . . . .	161
7.6.1	Classification vs. Regression . . . . .	161
7.6.2	Illustration and Interpretation of the Method . . . . .	161
7.6.3	Simulations: The Classification Case . . . . .	163
7.7	Conclusions . . . . .	165
<b>8</b>	<b>Applications and Case Studies</b>	<b>167</b>
8.1	System Identification with LS-SVMLab . . . . .	167
8.1.1	General Information . . . . .	167
8.1.2	Model Identification . . . . .	167
8.2	SYSID 2009: Wiener-Hammerstein Benchmark . . . . .	169
8.2.1	Model Structure . . . . .	169
8.2.2	Data Description and Training Procedure . . . . .	170
8.2.3	Estimation and Model Selection . . . . .	171
8.2.4	Results on Test Data . . . . .	174

---

8.3	Nonparametric Comparison of Densities Based on Statistical Bootstrap . . . . .	176
8.3.1	Introduction to the Problem . . . . .	176
8.3.2	Kernel Density Estimation . . . . .	177
8.3.3	Formulation and Construction of the Hypothesis Test . . . . .	180
8.3.4	Illustrative Examples . . . . .	183
8.4	Finding the Maximum in Hysteresis Curves . . . . .	185
8.5	Conclusions . . . . .	187
<b>9</b>	<b>Summary, Conclusions and Future Research</b>	<b>189</b>
9.1	Summary and Main Conclusions . . . . .	189
9.2	Future Research . . . . .	193
<b>A</b>	<b>Coupled Simulated Annealing</b>	<b>195</b>
	<b>References</b>	<b>197</b>
	<b>Curriculum vitae</b>	<b>221</b>

# Chapter 1

## Introduction

### 1.1 Historical Evolution and General Background

The regression estimation problem has a long history. Already in 1632 Galileo Galilei used a procedure which can be interpreted as fitting a linear relationship to contaminated observed data. Such fitting of a line through a cloud of points is the classical linear regression problem. Roughly 125 years later, Roger Joseph Boscovich (1757) addressed the fundamental mathematical problem of determining the parameters which best fits observational equations to data. Since then, a large number of estimation methods have been developed for linear regression. Four of the most commonly used methods are the least absolute deviations, least squares, trimmed least squares and M-regression.

Probably the most well-known method is the method of least squares, although Boscovich (1757) first considered least absolute deviations. The method of least squares was first published by Legendre in 1805 and by Gauss in 1809. Legendre and Gauss both applied the method to the problem of determining, from astronomical observations, the orbits of bodies around the sun. Gauss published a further development of the theory of least squares in 1821, including a version of the Gauss-Markov theorem. The term “regression” was coined by Francis Galton in the nineteenth century to describe a biological phenomenon. The phenomenon was that the heights of descendants of tall ancestors tend to regress down towards a normal average (a phenomenon also known as regression toward the mean). For Galton, regression had only this biological meaning, but his work was later extended by Yule (1897) and Pearson (1903) to a more general statistical context. In the work of Yule and Pearson, the joint distribution of the response and explanatory variables is assumed to be Gaussian. This assumption

was weakened by R.A. Fisher in his works of 1922 and 1925. Fisher assumed that the conditional distribution of the response variable is Gaussian, but the joint distribution need not be. In this respect, Fisher's assumption is closer to Gauss's formulation of 1821.

At some point in time, it became clear that it is not always easy to find a suitable parametric (linear or nonlinear) model to explain some phenomena. One was searching for a more flexible method where "the data would speak for themselves". For this reason, nonparametric smoothing methods were invented. Smoothing methods also have a long tradition. In the nineteenth century the nonparametric approach has been used as a major tool for empirical analysis: in 1857 the Saxonian economist Engel found the famous *Engelsches Gesetz* by constructing a curve which we would nowadays call a regressogram. The nonparametric smoothing approach has then long been neglected and the mathematical development of statistical theory in the first half of this century has mainly suggested a purely parametric approach for its simplicity in computation, its compatibility with model assumptions and also for its mathematical convenience.

The real breakthrough of these methods dates back to 1950s and early 1960s with pioneering articles of Rosenblatt (1956) and Parzen (1962) in the density estimation setting and with Nadaraya (1964) and Watson (1964) in the regression setting. Ever since, these methods are gaining more and more attention and popularity. Mainly, this is due to the fact that statisticians realized that pure parametric thinking in curve estimations often does not meet the need for flexibility in data analysis. Also the development of hardware created the demand for theory of now computable nonparametric estimates. However, nonparametric techniques have no intention of replacing parametric techniques. In fact, a combination of both can lead to the discovery of many interesting results which are difficult to accomplish by a single method e.g. semiparametric regression (Ruppert et al., 2003).

Regression methods continue to be an area of active research. In recent decades, new methods have been developed for robust regression, regression involving correlated responses such as time series and growth curves, regression in which the predictor or response variables are curves, images, graphs, or other complex data objects, regression methods accommodating various types of missing data, nonparametric regression, Bayesian methods for regression, regression in which the predictor variables are measured with error, regression with more predictor variables than observations, and causal inference with regression.

The increasing importance of regression estimation is also stimulated by the growth of information technology in the past twenty years. The demand for procedures capable of automatically extracting information from massive high-dimensional data sets is rapidly growing. Usually there is no prior knowledge available, leaving the data analyst with no other choice but a nonparametric approach. Often



these nonparametric techniques are pushed towards and possibly even over their limits because of their extreme flexibility. Therefore, caution is still advised when applying these techniques. Properties such as (universal) consistency (Stone, 1977) and rate of convergence may be not be neglected.

Kernel-based methodologies have been developed in the area of statistics (Rosenblatt, 1956; Nadaraya, 1964; Watson, 1964; Rao, 1983) and became popular in the fields of machine learning and data mining. The statistical learning framework proposed by Vapnik (Vapnik, 1999) led to the introduction of the Support Vector Machine (SVM) which has been successfully applied for nonlinear classification and regression in learning problems. The success of the SVM led to further developments in the area of kernel-based learning such as kernel principal component analysis (Jolliffe, 2002; Suykens et al., 2002), kernel canonical correlation analysis (Suykens et al., 2002) and kernel independent component analysis (Gretton et al., 2005). One particular class of kernel machines are the least squares support vector machines (LS-SVM) (Suykens and Vandewalle, 1999; Suykens et al., 2002) aimed at simplifying classical SVM formulations and developing a wider range of learning algorithms applicable beyond classification and regression. LS-SVMs are formulated using the  $L_2$  loss function in a constrained optimization framework with primal and dual formulations.

We developed the present LS-SVMLab toolbox version 1.7 (De Brabanter et al., 2010b) containing Matlab implementations for a number of LS-SVM algorithms. The toolbox and user's guide are freely available for research purposes and can be downloaded from <http://www.esat.kuleuven.be/sista/lssvmlab/>. The toolbox is mainly intended for use with the commercial Matlab package. The Matlab toolbox is compiled and tested for different computer architectures including Linux and Windows. Most functions can handle data sets up to 20.000 data points or more. LS-SVMLab's interface for Matlab consists of a basic version for beginners as well as a more advanced version with programs for multiclass encoding techniques. Also, a number of methods to estimate the generalization performance of the trained model (regression and classification) are included such as leave-one-out cross-validation,  $v$ -fold cross-validation and generalized cross-validation. The latest version also includes computation of pointwise and simultaneous confidence intervals for LS-SVM regression.

The amount of data available from fields such as bioinformatics, system identification and process industry is increasing at an explosive rate. Large-scale problems become more and more a challenge for supervised learning techniques in order to extract useful information from a "tsunami" of data. Approximation techniques, reduced set methods, subsampling schemes and sparse models are needed to deal with the large-scale data (up to 1.000.000 data points) using standard computers. The problem kernel methods face when dealing with large data sets is that the kernel matrix can become too large and hence memory problems often occur.

## 1.2 Practical Applications

Scientific data must be clean and reliable. While this can be the case in the majority of physical, chemical and engineering applications, biomedical data rarely possess such qualities. The very nature of biomedical objects is volatile and irregular, as are the results of biomedical assessments collected in large biomedical data sets. These data sets contain the results of tests which fluctuate with the patient's state, and long term trends are difficult to distinguish from short term fluctuations, taking into account that these data sets rarely contain reliable longitudinal components. The other typical problem is the large number of incomplete records, for example, if certain tests are missing for some individuals, then deleting such records may essentially reduce the power of the ongoing calculations. Even mortality statistics, probably the most reliable type of biomedical data, are not free from error: while the date of death is usually known precisely, the date of birth can be biased.

Next, we describe several applications in order to illustrate the practical relevance of nonparametric regression estimation. Notice that not all of the techniques/methods described in this thesis can be applied directly to these type of data sets.

### 1.2.1 Biomedical Data

**Example 1.1 (Survival Analysis)** *In survival analysis one is interested in predicting the survival time of a patient with a life-threatening disease given a description of the case, such as type of disease, blood measurements, sex, age, therapy, etc. The result can be used to determine the appropriate therapy for a patient by maximizing the predicted survival time with respect to the therapy (see Dippon et al., 2002) for an application in connection with breast cancer data). One specific feature in this application is that usually one cannot observe the survival time of a patient. Instead, one gets only the minimum of the survival time and a censoring time together with the information as to whether the survival time is less than the censoring time or not.*

**Example 1.2 (The Stanford Heart Transplant Program)** *Various versions of data from the transplant study has been reported in Fan and Gijbels (1994). The sample consisted of 157 cardiac patients who where enrolled in the transplantation program between October 1967 and February 1980. Patients alive beyond February 1980 were considered to be censored (55 in total). One of the questions of interest was the effect of the age of a patient receiving a heart transplantation, on his survival time after transplantation.*

## 1.2.2 Financial Data

**Example 1.3 (Interest Rate Data)** *Short-term risk-free interest rate play a fundamental role in financial markets. They are directly related to consumer spending, inflation and the overall economy.*

**Example 1.4 (Loan Management)** *A bank is interested in predicting the return on a loan given to a customer. Available to the bank is the profile of the customer including his credit history, assets, profession, income, age, etc. The predicted return affects the decision as to whether to issue or refuse a loan, as well as the conditions of the loan.*

**Example 1.5 (NYSE Data Set)** *The NYSE data set includes daily prices of 19 stocks out of 36 stocks of the NYSE old dataset along a 44-year period (11178 trading days) ending in 2006. From the 17 missing stocks: 13 companies became bankrupt after '85 and the other 4 stocks do not satisfy some liquidity constraints that can cause misleading results in the simulations. Further information can be found in Györfi et al. (2007), Györfi et al. (2008) and <http://www.cs.bme.hu/~oti/portfolio/>.*

## 1.2.3 System Identification

**Example 1.6 (Wiener-Hammerstein Benchmark)** *In such a structure there is no direct access to the static nonlinearity starting from the measured input-output, because it is sandwiched between two unknown dynamic systems. The signal-to-noise ratio of the measurements is quite high, which puts the focus of the benchmark on the ability to identify the nonlinear behavior, and not so much on the noise rejection properties (see e.g. De Brabanter et al., 2009). This application is described in more detail in Chapter 8.*

**Example 1.7 (Identification of a Pilot Scale Distillation Column)** *Input-output data was collected from a distillation column. The task is, given some control inputs, to identify the bottom and top temperature of the column. It is necessary to obtain accurate predictions of these temperatures, since they control the final quality of the product. More information can be found in Huyck et al. (2010). In this real life example, there is a linear model describing the measured temperature very accurately for both the top as well as the bottom temperature. However, a nonlinear ARX (obtained with least squares support vector machines) shows better performance compared to these best linear models (output-error, ARMAX, transfer function models). More information and practical considerations can be found in Chapter 8.*

### 1.2.4 Time Series Analysis

**Example 1.8 (Canadian Lynx Data)** *This data set consists of the annual fur returns of lynx at auction in London by the Hudson Bay Company for the period 1821-1934. This data reflects to some extent the population size of the lynx in the Mackenzie River district. It helps us to study the population dynamics of the ecological system in that area. Indeed, if the proportion of the number of lynx being caught to the population size remains approximately constant, after logarithmic transforms, the differences between the observed data and the population sizes remain approximately constant. For further background information on this data see Tong (1990).*

**Example 1.9 (Electric Load Forecasting)** *This particular data set consists of time series containing values of power load in 245 different high voltage - low voltage substations within the Belgian national grid operator ELIA for a period of 5 years. The sampling rate is 1 hour. The data characterize different profiles of load consumption such as business, residential and industrial. The yearly cycles are visible together with the daily cycles. From the daily cycle it is also possible to visualize morning, noon and evening peaks (Espinoza et al., 2006, 2007).*

## 1.3 Organization and Contributions of the Thesis

This thesis is organized in nine chapters. Figure 1.1 presents an overview of the chapters as well as their mutual relation.

### Chapter 2: Model Building

In Chapter 2 we give an overview of parametric and nonparametric modeling techniques. We introduce some measures of closeness which will be used in the rest of the thesis. Further, we emphasize the assumptions when one is using a certain model structure. We illustrate the fact that any universally consistent estimator can have an arbitrarily slow rate of convergence without imposing strong restrictions on the distribution of  $(X,Y)$ . Another important phenomenon is the curse of dimensionality. By means of a simple toy example, we show that the  $L_2$  distance between two random points uniformly distributed in a hypercube  $[0,1]^d$  will not go to zero even if the sample size is large. Finally, we discuss the four paradigms of nonparametric regression and give a short introduction to support vector machines (SVM) and least-squares support vector machines (LS-SVM). We also provide consistency results for some methods.

### Chapter 3: Model Selection Methods

From the methods introduced in Chapter 2, we know that most learning algorithms such as local polynomial regression, SVM, LS-SVM, etc. have some additional tuning parameters. In Chapter 3, we describe some commonly used methods (cross-validation and complexity criteria) to find suitable values for the tuning parameters in case of i.i.d. data and illustrate why such methods are necessary. In theory, one has to minimize the cost functions of these methods to find suitable parameters. This sometimes turns out to be a quite difficult task in practice due to the presence of many local minima. A standard method would define a grid over the parameters of interest and perform cross-validation for each of these grid values. However, three disadvantages come up with this approach: (i) the limitation of the desirable number of tuning parameters in a model, due to the combinatorial explosion of grid points. (ii) The practical inefficiency, namely, they are incapable of assuring the overall quality of the produced solution. (iii) Discretization fails to take into account the fact that the tuning parameters are continuous.

In order to overcome these drawbacks we propose a methodology consisting of two steps: first, determine good initial start values by means of a state of the art global optimization technique (coupled simulated annealing) and second, perform a fine-tuning derivative-free simplex search using the previous result as start value. Coupled simulated annealing accepts multiple coupled starters and is designed to easily escape from local optima and thus improves the quality of solution without compromising too much the speed of convergence. This will be the method of choice for finding the extra tuning parameters in the nonparametric regression models.

### Chapter 4: Fixed-Size Least Squares Support Vector Machines

Solving LS-SVMs for large data set is computationally and memory expensive due to the large size of the full kernel matrix. In order to handle large data sets (up to 1.000.000 data points on a standard PC), we introduce the so called fixed-size approach where a finite feature map can be approximated by the Nyström method. In this way, LS-SVM can be solved in the primal space and this turns out to be computationally less expensive than solving the dual. In order to construct a finite feature map one needs a number of representative points or vectors. These points or vectors are selected by means of maximizing the quadratic Rényi entropy. In addition, we show that the distribution of the selected sample of points or vectors is uniform over the input space.

This quadratic Rényi entropy criterion requires the tuning of an extra parameter (bandwidth of the kernel). Since large data sets are considered, one needs a fast and reliable method to select this parameter. Here, the bandwidth of the

kernel is determined via the solve-the-equation plug-in method. We employ a technique called improved fast Gauss transform to speed up the many summations of Gaussians needed in the solve-the-equation plug-in method. Further, we develop a fast and stable cross-validation procedure capably of handling large data sets. Finally, we demonstrate the ability of the proposed method on several benchmark data sets. The speed-up achieved by our algorithm is about 10 to 20 times compared to LIBSVM (state-of-the-art software for solving SVMs). We observed that our method requires less prototype vectors than support vectors in SVM, hence resulting into sparser models.

## Chapter 5: Robustness in Kernel Based Regression

The use of an  $L_2$  loss function and equality constraints for the models results into simpler formulations but on the other hand they have a potential drawback such as the lack of robustness. We discuss how one can robustify LS-SVM and FS-LSSVM via iteratively reweighting. In order to understand the robustness of these estimators against outliers, we use the empirical influence function and empirical maxbias curves. We showed that, in order to obtain a fully robust nonparametric method, three requirements have to be fulfilled i.e. (i) robust smoother, (ii) bounded kernel and (iii) a robust model selection procedure.

We compared four different weight functions and investigated their application in iteratively reweighted LS-SVM. We derived a weight function, Myriad, from the maximum likelihood estimation of a Cauchy distribution with a scaling factor. We showed that this weight function is very flexible and depending on the value of scale factor it can serve as a mean and mode estimator. Further, we illustrated the existence of a speed of convergence-robustness tradeoff. From these results, it follows that the proposed weight function is robust against extreme outliers but exhibits a slower speed of convergence compared to the other three weight functions.

By means of a toy example, we illustrate that the empirical maxbias of the proposed robust estimator increases very slightly with the number of outliers in region and stays bounded right up to the breakdown point. This is in strong contrast with the non-robust estimate which has a breakdown point equal to zero. Finally, the effectiveness of the proposed method is shown on a toy example and two real life data sets with extreme outliers.

## Chapter 6: Kernel Regression with Correlated Errors

In Chapter 6, we investigated the consequences when the i.i.d. (independent and identically distributed) assumption is violated. We showed that, for nonparametric kernel based regression, classical model selection procedures such as cross-

validation (CV), generalized CV and  $v$ -fold CV break down in the presence of correlated data and not the chosen smoothing method. Since the latter stays consistent when correlation is present in the data, it is not necessary to modify or add extra constraints to the smoother. We proved that by taking a kernel  $K$  satisfying  $K(0) = 0$ , the correlation structure is successfully removed without requiring any prior knowledge about its structure. By adding this extra constraint the kernel function  $K$  it implies that  $K$  cannot be unimodal. We also show that there exist no optimal class of kernels satisfying the constraints of our theorem. Further, we showed both theoretically and experimentally, that by using bimodal kernels the estimate will suffer from increased mean squared error. In order to reduce both effects on the estimation, we developed a class of so-called  $\varepsilon$ -optimal class of bimodal kernels which approaches the optimal kernel for  $\varepsilon \rightarrow 0$ .

Since we proved that a bimodal kernel can never positive (semi) definite, we developed a model selection procedure for LS-SVM in order to effectively handle correlation in the data. Finally, we illustrated the proposed method on toy examples with different noise models.

## Chapter 7: Confidence and Prediction Intervals

We discussed the construction of bias-corrected  $100(1 - \alpha)\%$  approximate confidence and prediction intervals (pointwise and uniform) for linear smoothers, in particular for LS-SVM. We proved, under certain conditions, the asymptotic normality of LS-SVM. In order to estimate the bias without estimating higher order derivatives, we discussed a technique called double smoothing. Further, we developed a nonparametric variance estimator which can be related to other well-known nonparametric variance estimators.

In order to obtain uniform or simultaneous confidence intervals we used two techniques i.e Bonferroni/Šidák correction and volume-of-tube formula. We provided extensions of this formula in higher dimensions and discussed how to compute some of the coefficients in practice. We illustrated that the width of the bands are expanding with increasing dimensionality by means of an example. By means of a Monte Carlo study, we demonstrated that the proposed bias-corrected  $100(1 - \alpha)\%$  approximate simultaneous confidence intervals achieve the proper empirical coverage rate. By comparing our proposed method with bootstrap based simultaneous confidence interval, we concluded that both methods produced similar intervals.

Since classification and regression are equivalent for LS-SVM, the proposed simultaneous confidence intervals are extended to the classification case. Finally, we provided graphical illustrations and interpretations of the proposed method for classification.

## **Chapter 8: Applications and Case Studies**

We discuss some practical examples and case studies. We demonstrate the capabilities of the developed techniques in several scientific domains. First, we show that FS-LSSVM is a powerful tool for black-box modeling and is capable of handling large data sets. Second, by transforming LS-SVM for regression to a density estimator via a binning technique, we formulate a hypothesis test based on bootstrap with variance stabilization. This test can assist the researcher to decide which user specified parametric distribution best fits the given data. Finally, we use LS-SVM to determine the maximum shift in hysteresis curves.

## **Chapter 9: Summary, Conclusions and Future Research**

In the last Chapter of this thesis, summary and conclusions of the presented studies are given chapter by chapter. We also discuss some suggestions for further research.



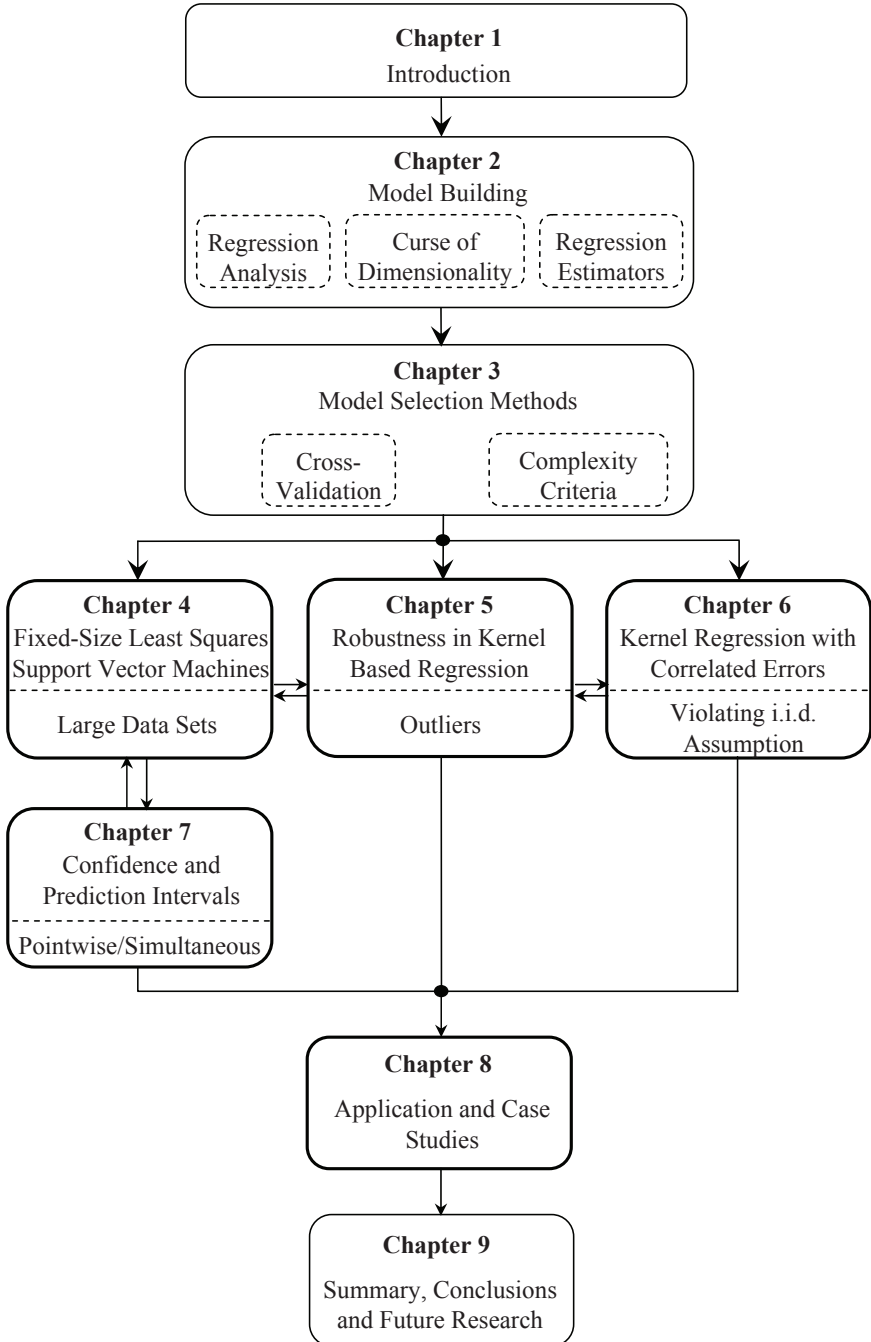


Figure 1.1: Overview and mutual relation between different chapters in the thesis.



# Chapter 2

## Model Building

In this Chapter, we give an overview of various ways to define parametric and nonparametric regression estimates. First, we introduce some measure of closeness and derive the regression function that minimizes the  $L_2$  risk. Second, we discuss the assumptions and restrictions on the regression function and clarify that any regression estimate can have an arbitrary slow rate of convergence. Finally, we illustrate the curse of dimensionality by means of an example and by giving an overview of parametric and nonparametric modeling techniques.

### 2.1 Regression Analysis and Loss Functions

A model is just an abstraction of reality and it provides an approximation of some relatively more complex phenomenon. Models may be broadly classified as deterministic or probabilistic. Deterministic models abound in science and engineering e.g. Ohm's law, the ideal gas law and the laws of thermodynamics. An important task in statistics is to find a probabilistic model, if any, that exist in a set of variables being subject to random fluctuations and possibly measurement error. In regression problems typically one of the variables, often called the response, output, observation or dependent variable, is of particular interest. The other variables, usually called explanatory, input, covariates, regressor or independent variables, are primarily used to explain the behavior of the response variable.

Let  $X \in \mathcal{X} \subseteq \mathbb{R}^d$  denote a real valued random input vector and  $Y \in \mathcal{Y} \subseteq \mathbb{R}$  a real valued output variable with joint distribution  $F_{XY}$ . In regression analysis one is interested in how the value of the response variable  $Y$  depends on the value of the observation vector  $X$ . In fact, one wants to find a (measurable) function  $f : \mathbb{R}^d \rightarrow \mathbb{R}$ , such that  $f(X)$  is a *good approximation of  $Y$*  or in other words

$|f(X) - Y|$  should be made small. However, since  $X$  and  $Y$  are random it is not really clear what *small*  $|f(X) - Y|$  means. To overcome this problem, it is common to introduce so-called risk or loss functions e.g.  $L_2$  risk or mean squared error (MSE)

$$\mathcal{R}(f) = \mathbf{E}|f(X) - Y|^2,$$

where  $\mathbf{E}[X] = \int_{\mathbb{R}^d} x dF(x)$  with  $F$  the distribution of  $X$ . Then, the latter is required to be as small as possible.

While it is quite natural to use the expectation operator, it is not obvious why one wants to minimize the expectation of the squared distance between  $f(X)$  and  $Y$  and not, more generally, the  $L_p$  risk

$$\mathbf{E}|f(X) - Y|^p$$

for some  $p \geq 1$ . There are two reasons for considering the  $L_2$  risk or loss. First, the mathematical treatment of the whole problem is simplified. As shown below, the function that minimizes the  $L_2$  risk can be derived explicitly. Second, minimizing the  $L_2$  risk leads naturally to estimates which can be computed rapidly e.g. the ordinary least squares problem.

We are interested in a (measurable) function  $m^* : \mathcal{X} \rightarrow \mathcal{Y}$  such that

$$\mathbf{E}|m^*(X) - Y|^2 = \min_{f: \mathcal{X} \rightarrow \mathcal{Y}} \mathbf{E}|f(X) - Y|^2.$$

Such a function can be explicitly found as follows. Let

$$m(x) = \mathbf{E}[Y|X = x]$$

be the regression function. Then, the regression function minimizes the  $L_2$  risk. For any arbitrary  $f : \mathcal{X} \rightarrow \mathcal{Y}$ , one has

$$\begin{aligned} \mathbf{E}|f(X) - Y|^2 &= \mathbf{E}|f(X) - m(X) + m(X) - Y|^2 \\ &= \mathbf{E}|f(X) - m(X)|^2 + \mathbf{E}|m(X) - Y|^2, \end{aligned}$$

because

$$\begin{aligned} \mathbf{E}[(f(X) - m(X))(m(X) - Y)] &= \mathbf{E}[\mathbf{E}[(f(X) - m(X))(m(X) - Y)|X]] \\ &= \mathbf{E}[(f(X) - m(X))\mathbf{E}[(m(X) - Y)|X]] \\ &= \mathbf{E}[(f(X) - m(X))(m(X) - \mathbf{E}[Y|X])] \\ &= \mathbf{E}[(f(X) - m(X))(m(X) - m(X))] \\ &= 0. \end{aligned}$$

Hence,

$$\mathbf{E} |f(X) - Y|^2 = \int_{\mathbb{R}^d} |f(x) - m(x)|^2 dF(x) + \mathbf{E} |m(X) - Y|^2, \quad (2.1)$$

where  $F$  denotes the distribution of  $X$ . The first term is always nonnegative and is zero when  $f(x) = m(x)$ . Therefore,  $m^*(x) = m(x)$  i.e. the optimal approximation (with respect to the  $L_2$  risk) of  $Y$  by a function of  $X$  is given by  $m(X)$ .

In applications, the distribution of  $(X, Y)$  and hence the regression function is usually unknown. Therefore, it is impossible to predict  $Y$  using  $m(X)$ . But on the other hand, it is often possible to observe data according to the distribution of  $(X, Y)$  and to estimate the regression function from data. In the regression estimation setting, one wants to use the data  $\mathcal{D}_n = \{(X_1, Y_1), \dots, (X_n, Y_n)\}$ , where  $(X_1, Y_1), \dots, (X_n, Y_n)$  are independent and identically distributed (i.i.d.) random variables with  $\mathbf{E}[Y^2] < \infty$ , in order to construct an estimate  $\hat{m}_n : \mathcal{X} \rightarrow \mathcal{Y}$  of the regression function  $m$ . Since in general estimates will not be equal to the regression function, one needs error criteria which measure the difference between the regression function and an arbitrary estimate. First, the pointwise error,

$$|\hat{m}_n(x) - m(x)| \quad \text{for some fixed } x \in \mathbb{R}^d,$$

second, the supremum norm error,

$$\|\hat{m}_n(x) - m(x)\|_\infty = \sup_{x \in C} |\hat{m}_n(x) - m(x)| \quad \text{for some fixed set } C \subseteq \mathbb{R}^d,$$

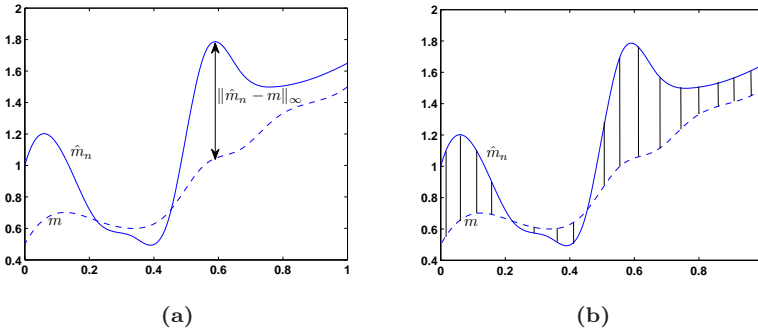
and third, the  $L_p$  error,

$$\int_C |\hat{m}_n(x) - m(x)|^p dx,$$

where the integration is with respect to the Lebesgue measure,  $C$  is a fixed subset of  $\mathbb{R}^d$  and  $p \geq 1$  is arbitrary. In many cases  $p = 2$  is often used. Other error criteria are also possible such as the Hellinger distance, Kullback-Leibler, . . . . Further theoretical aspects of each of these error criteria as well as their mutual relations can be found in Halmos (1974) and Tsybakov (2009). As an example, let  $[a, b] \subset \mathbb{R}$  be a non-empty closed and bounded interval and let  $\hat{m}_n$  and  $m \in C[a, b]$ . Figure 2.1 graphically illustrates the supremum norm error (Figure 2.1a) and the  $L_1$  error norm (Figure 2.1b).

Recall that the main goal was to find a function  $f$  such that the  $L_2$  risk  $\mathbf{E} |f(X) - Y|^2$  is small. The minimal value of the  $L_2$  risk is  $\mathbf{E} |m(X) - Y|^2$  and it is achieved by the regression function  $m$ . Similarly to (2.1), one can show that the  $L_2$  risk  $\mathbf{E}[|\hat{m}_n(X) - Y|^2 | \mathcal{D}_n]$  of an estimate  $\hat{m}_n$  satisfies

$$\mathbf{E}[|\hat{m}_n(X) - Y|^2 | \mathcal{D}_n] = \int_{\mathbb{R}^d} |\hat{m}_n(x) - m(x)|^2 dF(x) + \mathbf{E} |m(X) - Y|^2.$$



**Figure 2.1:** (a) Supremum norm error. The vertical line shows the largest distance between  $\hat{m}_n$  and  $m$ ; (b)  $L_1$  norm error. This is the area between  $\hat{m}_n$  and  $m$ .

Thus the  $L_2$  risk of an estimate  $\hat{m}_n$  is close to the optimal value if and only if the  $L_2$  error

$$\int_{\mathbb{R}^d} |\hat{m}_n(x) - m(x)|^2 dF(x)$$

is close to zero.

## 2.2 Assumptions, Restrictions and Slow Rate

From the previous Section we know that regression function  $m$  satisfies

$$\mathbf{E} |m(X) - Y|^2 = \min_{f: \mathcal{X} \rightarrow \mathcal{Y}} \mathbf{E} |f(X) - Y|^2.$$

Here, the minimum is taken over all measurable functions  $f : \mathcal{X} \rightarrow \mathcal{Y}$ . This is impossible in the regression estimation problem since the risk functional depends on the distribution of  $(X, Y)$  which is usually unknown in practice. Given the observations  $X$  and the training data  $\mathcal{D}_n = \{(X_1, Y_1), \dots, (X_n, Y_n)\}$  of i.i.d random variables, minimizing the empirical  $L_2$  risk functional defined as

$$\mathcal{R}_{\text{emp}}(f) = \frac{1}{n} \sum_{k=1}^n (f(X_k) - Y_k)^2 \quad (2.2)$$

leads to infinitely many solutions. Indeed, any function  $\hat{f}_n$  passing through the training data  $\mathcal{D}_n$  is a solution. To obtain useful results for finite number of points, one must *restrict* the solution to (2.2) to a smaller set of functions. First, choose a class of suitable functions  $\mathcal{F}_n$  and select a function  $f : \mathcal{X} \rightarrow \mathcal{Y}$ , where  $f \in \mathcal{F}_n$  is

the minimizer of the  $L_2$  risk functional. The estimate  $\hat{m}_n$  is defined as follows

$$\hat{m}_n \in \mathcal{F}_n \quad \text{and} \quad \frac{1}{n} \sum_{k=1}^n (\hat{m}_n(X_k) - Y_k)^2 = \min_{f \in \mathcal{F}_n} \frac{1}{n} \sum_{k=1}^n (f(X_k) - Y_k)^2.$$

A possible model structure states that the training data  $\mathcal{D}_n = \{(X_1, Y_1), \dots, (X_n, Y_n)\}$  of i.i.d random variables can be written as

$$Y_k = m(X_k) + e_k. \tag{2.3}$$

In (2.3) one *assumes* that the error term  $e$  in the model has zero mean and constant variance i.e.  $\mathbf{E}[e_k | X = x_k] = 0$  and  $\mathbf{E}[e_k^2 | X = x_k] = \sigma^2 < \infty$ , and  $\{e_k\}$  are uncorrelated random variables i.e.  $\mathbf{Cov}[e_k, e_j] = 0$  for  $k \neq j$ . Further, one *assumes* that the observations  $X_1, \dots, X_n$  could be measured accurately. Otherwise, if the observations  $X_1, \dots, X_n$  are measured with error, the true values would be unknown and a errors-in-variables model (linear or nonlinear) is needed (Fuller, 1987; Carroll et al., 2006). More recent work, in the nonparametric setting, regarding the topic of measurement errors as well as deconvolution can be found in Meister (2009).

For most systems, the pairs  $(X, Y)$  will not have a deterministic relationship i.e.  $Y_k = m(X_k)$ . Generally, there will be other unmeasurable variables that will also contribute to  $Y$  including measurement errors (the measurement error is strictly on the  $Y$ 's and not on the  $X$ 's). The “additive error model” (2.3) *assumes* that one can capture all these deviations from a deterministic relationship via the error  $e$ .

Consider the  $L_2$  error criterion

$$\|\hat{m}_n - m\|^2 = \int (\hat{m}_n(x) - m(x))^2 dF(x).$$

The average  $L_2$  error,  $\mathbf{E} \|\hat{m}_n - m\|^2$ , is completely determined by the distribution of  $(X, Y)$  and the regression function estimator  $\hat{m}_n$ . Although there exist universally consistent regression estimates e.g. neural networks estimates, kernel estimates, . . . one would be interested in regression function estimates with  $\mathbf{E} \|\hat{m}_n - m\|^2$  tending to zero with a “guaranteed” rate of convergence. Disappointingly, such estimates do not exist. Györfi et al. (2002, Chapter 3) have shown that is impossible to obtain nontrivial rate of convergence results without imposing strong *restrictions* on the distribution of  $(X, Y)$ . Even when the distribution of  $X$  is good and  $Y = m(X)$ , the rate of convergence of “any” estimator can be arbitrary slow (Györfi et al., 2002; Devroye et al., 2003). For completeness, we state this strong result in the following theorem.

**Theorem 2.1 (Györfi et al., 2002)** *Let  $\{a_n\}$  be a sequence of positive numbers converging to zero. For every sequence of regression estimates, there exists a*

distribution of  $(X, Y)$ , such that  $X$  is uniformly distributed on  $[0, 1]$ ,  $Y = m(X)$ ,  $m$  is  $\pm 1$  valued, and

$$\limsup_{n \rightarrow \infty} \frac{\mathbf{E} \|\hat{m}_n - m\|^2}{a_n} \geq 1.$$

Therefore, rate of convergence studies for particular estimates must necessarily be accompanied by conditions on  $(X, Y)$ . Only under certain regularity conditions it is possible to obtain upper bounds for the rate of convergence to zero for  $\mathbf{E} \|\hat{m}_n - m\|^2$ . One can derive the optimal rate of convergence for a certain class of distributions of  $(X, Y)$  by imposing some smoothness conditions on the regression function depending on some parameter  $p$ :

**Definition 2.1** Let  $p = k + \beta$  for some  $k \in \mathbb{N}_0$  and  $0 < \beta \leq 1$ , and let  $C > 0$ . A function  $f : \mathbb{R}^d \rightarrow \mathbb{R}$  is called  $(p, C)$ -smooth if for every  $v = (v_1, \dots, v_n)$ ,  $v_i \in \mathbb{N}_0$ ,  $\sum_{j=1}^d v_j = k$  the partial derivative  $\frac{\partial^k f}{\partial x_1^{v_1} \dots \partial x_d^{v_d}}$  exists and satisfies

$$\left| \frac{\partial^k f}{\partial x_1^{v_1} \dots \partial x_d^{v_d}}(x) - \frac{\partial^k f}{\partial x_1^{v_1} \dots \partial x_d^{v_d}}(z) \right| \leq C \cdot \|x - z\|^\beta \quad (x, z \in \mathbb{R}^d).$$

For classes  $\mathcal{F}_{n,p}$ , where  $m$  is  $p$  times continuously differentiable, the optimal rate of convergence will be  $n^{-\frac{2p}{2p+d}}$  (Györfi et al., 2002). This optimal rate shows, if we consider  $p$  fixed, a clear dependence on the dimensionality of the problem i.e. the dimension of the design variable  $X$ . We will clarify this in more detail in the next Section.

## 2.3 Curse of Dimensionality

From the previous Section we know that estimating a regression function is more and more difficult if the dimensionality of the problem is becoming larger. This phenomenon is commonly referred to as the *curse of dimensionality* and was first reported by Bellman (1961) in the context of approximation theory to signify the fact that estimation difficulty not only increases with dimension but can also increase superlinearly. The reason for this is that in the case of large  $d$  it is, in general, not possible to densely pack the space of  $X$  with finitely many sample points, even if the sample size  $n$  is very large (see also Kendall (1961) for a study of the geometry in higher dimensions). we will illustrate this by means of an example.

Let  $X_1, \dots, X_n$  be i.i.d.  $\mathbb{R}^d$ -valued random variables with  $X$  uniformly distributed in the hypercube  $[0, 1]^d$ . Denote the expected  $L_2$ -norm distance of  $X$  to its nearest neighbor in  $X_1, \dots, X_n$  by  $d_2(d, n)$ , i.e. set

$$d_2(d, n) = \mathbf{E} \left[ \min_{i=1, \dots, n} \|X - X_i\| \right].$$



Here  $\|x\|$  is the  $L_2$  norm of a vector  $x = (x^{(1)}, \dots, x^{(d)})^T \in \mathbb{R}^d$  defined by

$$\|x\| = \sqrt{\sum_{i=1}^d (x^{(i)})^2}.$$

Then

$$\begin{aligned} d_2(d,n) &= \int_0^\infty \mathbf{P} \left[ \min_{i=1, \dots, n} \|X - X_i\| > t \right] dt \\ &= \int_0^\infty \left( 1 - \mathbf{P} \left[ \min_{i=1, \dots, n} \|X - X_i\| \leq t \right] \right) dt. \end{aligned}$$

The bound

$$\begin{aligned} \mathbf{P} \left[ \min_{i=1, \dots, n} \|X - X_i\| \leq t \right] &\leq n \mathbf{P} [\|X - X_1\| \leq t] \\ &\leq n \frac{\pi^{d/2}}{\Gamma(\frac{d}{2} + 1)} t^d, \end{aligned}$$

where  $\frac{\pi^{d/2}}{\Gamma(\frac{d}{2} + 1)} t^d$  is the volume of a ball in  $\mathbb{R}^d$  with radius  $t$  and  $\Gamma(x) = \int_0^\infty t^{x-1} e^{-t} dt$  for  $x > 0$  satisfies  $\Gamma(x + 1) = x\Gamma(x)$ ,  $\Gamma(1) = 1$  and  $\Gamma(1/2) = \sqrt{\pi}$ , implies

$$\begin{aligned} d_2(d,n) &\geq \int_0^{\frac{\Gamma(\frac{d}{2} + 1)^{1/d}}{\sqrt{\pi n^{1/d}}}} \left( 1 - n \frac{\pi^{d/2}}{\Gamma(\frac{d}{2} + 1)} t^d \right) dt \\ &= \left[ t - \frac{n}{d+1} \frac{\pi^{d/2}}{\Gamma(\frac{d}{2} + 1)} t^{d+1} \right]_0^{\frac{\Gamma(\frac{d}{2} + 1)^{1/d}}{\sqrt{\pi n^{1/d}}}} \\ &= \frac{\Gamma(\frac{d}{2} + 1)^{1/d}}{\sqrt{\pi n^{1/d}}} - \frac{n}{d+1} \frac{\pi^{d/2}}{\Gamma(\frac{d}{2} + 1)} \frac{\Gamma(\frac{d}{2} + 1)^{1+\frac{1}{d}}}{n^{1+\frac{1}{d}} \pi^{\frac{d+1}{2}}} \\ &= \frac{d}{d+1} \frac{\Gamma(\frac{d}{2} + 1)^{1/d}}{\sqrt{\pi}} \frac{1}{n^{1/d}}. \end{aligned}$$

Table 2.1 shows values of this lower bound for various values of  $d$  and  $n$ . It is clear that for dimension  $d = 10$  and larger this lower bound is not close to zero even if the sample size  $n$  is large. So for most values of  $x$  one only has data points  $(X_i, Y_i)$  available where  $X_i$  is not close to  $x$ . At such data points  $m(X_i)$  will, in

**Table 2.1:** Lower bounds for  $d_2(d,n)$  for i.i.d.  $\mathbb{R}^d$ -valued random variables with  $X$  uniformly distributed in the hypercube  $[0,1]^d$

	$n = 100$	$n = 1000$	$n = 10.000$	$n = 100.000$
$d_2(1,n)$	$\geq 0.0025$	$\geq 0.00025$	$\geq 0.000025$	$\geq 0.0000025$
$d_2(10,n)$	$\geq 0.5223$	$\geq 0.4149$	$\geq 0.3296$	$\geq 0.2618$
$d_2(20,n)$	$\geq 0.9083$	$\geq 0.8095$	$\geq 0.7215$	$\geq 0.6430$
$d_2(50,n)$	$\geq 1.6093$	$\geq 1.5369$	$\geq 1.4677$	$\geq 1.4016$

general, not be close to  $m(x)$  even for a smooth regression function. Naturally, a similar phenomenon also occurs if one replaces the  $L_2$  norm by the supremum norm. Notice that the above arguments are no longer valid if the components of  $X$  are not independent.

So we observed that the optimal rate of convergence converges to zero rather slowly if the dimension  $d$  is large compared to  $p$  (degree of smoothness). Vapnik (1999) has shown that the asymptotic rate of convergence decreases with increasing input dimension when the characteristic of smoothness remains fixed. Therefore, one can guarantee good estimation of a high dimensional function only if the function  $m \in \mathcal{F}_p$  for  $p \rightarrow \infty$  i.e.  $m$  is extremely smooth.

In practice, one can circumvent the curse of dimensionality by posing additional assumptions on the regression function. Examples of such techniques are additive models (Breiman and Friedman, 1985; Buja et al., 1989; Hastie and Tibshirani, 1990), projection pursuit (Friedman and Tukey, 1974; Friedman and Stuetzle, 1981) and tree-based methods (Breiman et al., 1984) and its variants (Friedman, 1991; Jordan and Jacobs, 1994). More recently, Ferraty and Vieu (2006) showed that the curse of dimensionality does not affect functional data with high correlation but is dramatic for the uncorrelated ones. Nevertheless, by considering functional features, even if the data are not correlated, the curse of dimensionality can be partially canceled out.

We conclude this Section with a remarkable quote from Clarke et al. (2009) regarding the curse of dimensionality

*“Suppose you had to use an estimator to estimate  $m(x)$  when  $d = 20.000$  and data collection was not rapid. Then, humans could well have evolved into a different species rendering the analysis meaningless, before the estimator got close to the true function.”*

## 2.4 Parametric and Nonparametric Regression Estimators: An Overview

In this Section we give an overview of various ways to define parametric and nonparametric estimates. The description of the four paradigms of nonparametric regression is based on Friedman (1991), Fan and Gijbels (1996), Györfi et al. (2002) and Hastie et al. (2009).

### 2.4.1 Parametric Modeling

The classical approach for estimating a regression function is to use parametric regression estimation. One assumes that the structure of the regression function is known and depends only on finitely many parameters. The linear regression model provides a relatively flexible framework. However, linear regression models are not appropriate for all situations. There are many situations where the dependent variable and the independent variables are related through a known nonlinear function. It should be clear that in dealing with the linear and nonlinear regression models the normal distribution plays a central role. There are a lot of practical situations where this assumption is not going to be even approximately satisfied. Extensions to this classical linear model are also developed and are called generalized linear models, see e.g. McCullagh and Nelder (1999) for a detailed description.

Consider, as an example, linear regression estimation. Let  $\mathcal{F}_n$  denote a class of linear combinations of the components of  $X = (X^{(1)}, \dots, X^{(d)})^T \in \mathbb{R}^d$

$$\mathcal{F}_n = \left\{ m : m(x) = \beta_0 + \sum_{i=1}^d \beta_i X^{(i)}, \quad \beta_i \in \mathbb{R} \right\}.$$

The unknown parameters  $\beta_0, \dots, \beta_d$  can be estimated from the data  $\mathcal{D}_n = \{(X_1, Y_1), \dots, (X_n, Y_n)\}$  e.g. by applying the principle of least squares

$$(\hat{\beta}_0, \dots, \hat{\beta}_d) = \arg \min_{\beta_0, \dots, \beta_d \in \mathbb{R}^d} \left[ \frac{1}{n} \sum_{k=1}^n \left( Y_k - \beta_0 - \sum_{i=1}^d \beta_i X_k^{(i)} \right)^2 \right].$$

The estimate  $\hat{m}_n$  can be evaluated in a point  $x = (x^{(1)}, \dots, x^{(d)})^T \in \mathbb{R}^d$  leading to

$$\hat{m}_n(x) = \hat{\beta}_0 + \sum_{i=1}^d \hat{\beta}_i x^{(i)}.$$

However, parametric estimates have a drawback. Regardless of the data, a parametric estimate cannot approximate the regression function better than the

best function with the assumed parametric structure. This inflexibility concerning the structure of the regression function is avoided by nonparametric regression estimates.

**Remark** *If  $(X, Y)$  is jointly Gaussian, then  $m(x)$  is a linear function.*

## 2.4.2 Local Averaging

### Nadaraya-Watson kernel smoother

As a first example of a local averaging estimate consider the Nadaraya-Watson (NW) kernel smoother which was independently proposed by Nadaraya (1964) and Watson (1964). Recall from Section 2.1 that the regression function is given by

$$\begin{aligned} m(x) &= \mathbf{E}[Y|X = x] = \int y f_{Y|X}(y|x) dy \\ &= \frac{1}{f_X(x)} \int y f_{X,Y}(x,y) dy, \end{aligned} \quad (2.4)$$

where  $f_X(x)$ ,  $f_{X,Y}(x,y)$  and  $f_{Y|X}(y|x)$  denote the marginal density of  $X$ , the joint density of  $X$  and  $Y$  and the conditional density of  $Y$  given  $X$  respectively. Let  $K^* : \mathbb{R} \rightarrow \mathbb{R}$  and  $K : \mathbb{R}^d \rightarrow \mathbb{R}$  be isotropic kernel functions, i.e. the argument of the kernel only depends on the distance between two points and not on multiplications between points such as the linear or polynomial kernel, and let  $h, h^* > 0$  denote the bandwidths or smoothing parameters. Then, the unknown quantity  $f_{X,Y}(x,y)$  in (2.4) can be estimated by a bivariate kernel estimate with a product kernel (Silverman, 1986; Scott, 1992)

$$\hat{f}_{X,Y}(x,y) = \frac{1}{nh^d h^*} \sum_{i=1}^n K\left(\frac{x - X_i}{h}\right) K^*\left(\frac{y - Y_i}{h^*}\right).$$

The quantity  $\int y f_{X,Y}(x,y) dy$  can be estimated by

$$\begin{aligned} \int y \hat{f}_{X,Y}(x,y) dy &= \frac{1}{nh^d h^*} \sum_{i=1}^n \int y K\left(\frac{x - X_i}{h}\right) K^*\left(\frac{y - Y_i}{h^*}\right) dy \\ &= \frac{1}{nh^d} \sum_{i=1}^n K\left(\frac{x - X_i}{h}\right) \frac{1}{h^*} \int y K^*\left(\frac{y - Y_i}{h^*}\right) dy \\ &= \frac{1}{nh^d} \sum_{i=1}^n K\left(\frac{x - X_i}{h}\right) \int (Y_i + uh^*) K^*(u) du \\ &= \frac{1}{nh^d} \sum_{i=1}^n K\left(\frac{x - X_i}{h}\right) Y_i, \end{aligned}$$

if  $\int K(u) du = 1$  and  $\int uK(u) du = 0$ . These two conditions are fulfilled when  $K$  is a symmetric probability density function. Since an estimator for the marginal density of  $X$  is given by

$$\hat{f}_X(x) = \frac{1}{nh^d} \sum_{i=1}^n K\left(\frac{x - X_i}{h}\right),$$

the resulting estimator for  $m(x)$  is the Nadaraya-Watson kernel smoother

$$\hat{m}_n(x) = \sum_{i=1}^n \frac{K\left(\frac{x-X_i}{h}\right) Y_i}{\sum_{i=1}^n K\left(\frac{x-X_i}{h}\right)}. \quad (2.5)$$

Notice that (2.5) has universally consistency properties even in the case when  $X$  and  $Y$  have no densities. The NW kernel smoother is most natural in the random design case. Indeed, if the marginal density  $f_X(x)$  is known, one can use this instead of  $\hat{f}_X(x)$  in (2.5). Then the following estimator, which is slightly different from the NW kernel smoother, is obtained

$$\hat{m}_n(x) = \frac{1}{nh^d f_X(x)} \sum_{i=1}^n K\left(\frac{x - X_i}{h}\right) Y_i.$$

In particular, if  $f_X$  is the density of the uniform distribution on  $[0,1]$ , then

$$\hat{m}_n(x) = \frac{1}{nh^d} \sum_{i=1}^n K\left(\frac{x - X_i}{h}\right) Y_i. \quad (2.6)$$

There exists a vast literature studying these type of estimators for fixed design e.g. the Priestley-Chao estimator (Priestley and Chao, 1972) and the Gasser-Müller estimator (Gasser and Müller, 1979) are closely related to (2.6).

If one uses the naive kernel (or window kernel)  $K(u) = I_{\{\|u\| \leq 1\}}$ , then (2.5) yields

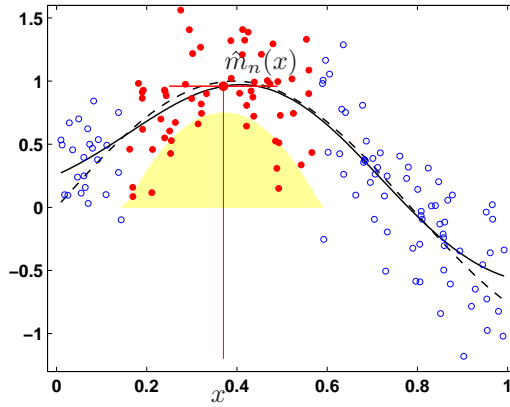
$$\hat{m}_n(x) = \sum_{i=1}^n \frac{I_{\{\|x-X_i\| \leq h\}} Y_i}{\sum_{i=1}^n I_{\{\|x-X_i\| \leq h\}}},$$

i.e. one estimates  $m(x)$  by averaging  $Y_i$ 's such that the distance between  $X_i$  and  $x$  is not larger than  $h$ . When considering multivariate data, simply replace  $K(u)$  by  $K(\|u\|)$ .

Notice that (2.5) can be considered as locally fitting a constant to the data. In fact (2.5) satisfies

$$\hat{m}_n(x) = \arg \min_{c \in \mathbb{R}} \frac{1}{n} \sum_{i=1}^n K\left(\frac{x - X_i}{h}\right) (Y_i - c)^2. \quad (2.7)$$

From the above formulation for the NW kernel smoother it is clear that it corresponds to locally approximating the regression function with a constant where the weight of the  $Y_i$  depends on the distance between  $X_i$  and  $x$ . Figure 2.2 illustrates the principle of the Nadaraya-Watson kernel smoother for the one dimensional case.



**Figure 2.2:** 100 pairs  $(X_i, Y_i)$  are generated at random from  $Y = \sin(4X)$  (dashed line) with Gaussian errors  $\varepsilon \sim \mathcal{N}(0, 1/3)$  and  $X \sim \mathcal{U}[0, 1]$ . The dot around 0.38 (vertical line) is the fitted constant  $\hat{m}_n(x)$ , and the full circles indicate those observations contributing to the fit at  $x$ . The solid region indicates the weights assigned to observations according to the Epanechnikov kernel. The full NW estimate is shown by the full line.

### **$k$ -Nearest Neighbor Regression Estimate**

As a second example of local averaging consider the  $k$ -nearest neighbor estimate. Here one determines the  $k$  nearest  $X_i$ 's to  $x$  in terms of distance  $\|x - X_i\|$  and estimates  $m(x)$  by averaging the corresponding  $Y_i$ 's. For  $X \in \mathbb{R}^d$ , let  $(X_{(1)}(x), Y_{(1)}(x)), \dots, (X_{(n)}(x), Y_{(n)}(x))$  be a permutation of  $(X_1, Y_1), \dots, (X_n, Y_n)$  such that

$$\|x - X_{(1)}(x)\| \leq \dots \leq \|x - X_{(n)}(x)\|.$$

The  $k$ -nearest neighbor estimate is defined by

$$\hat{m}_n(x) = \frac{1}{k} \sum_{i=1}^k Y_{(i)}(x).$$

Here the weight equals  $\frac{1}{k}$  if  $X_i$  is among the  $k$  nearest neighbors of  $x$  and equals zero otherwise.

Concerning the consistency of the  $k$ -nearest neighbor regression estimate Devroye et al. (1994) presented the following two results. First, all modes of convergence in  $L_1$  (in probability, almost sure, complete) are equivalent if for all distributions  $(X, Y)$  the regression variable is bounded. Further, by the boundedness of  $Y$  they also obtained  $L_p$ -consistency. Second, if  $k$  is chosen to satisfy  $\lim_{n \rightarrow \infty} k / \log(n) = \infty$  and  $\lim_{n \rightarrow \infty} k/n = \infty$  strong universal consistency of the estimate was obtained even if  $Y$  is not bounded. The estimate has sense and universal properties even when the density does not exist.

### 2.4.3 Local Modeling

A generalization of (2.7) leads to this class of modeling i.e. instead of locally fitting a constant to the data, locally fit a more general function which depends on several parameters. Let  $g(\cdot, \{a_k\}_{k=1}^l) : \mathbb{R}^d \rightarrow \mathbb{R}$  be a function depending on parameters  $\{a_k\}_{k=1}^l$ . For each  $x \in \mathbb{R}^d$ , choose values of these parameters by a local least squares criterion

$$\{\hat{a}_k\}_{k=1}^l = \arg \min_{\{a_k\}_{k=1}^l} \frac{1}{n} \sum_{i=1}^n K \left( \frac{x - X_i}{h} \right) (Y_i - g(X_i, \{a_k\}_{k=1}^l))^2. \quad (2.8)$$

Notice that if one chooses  $g(x, \{a_k\}_{k=1}^l) = c$  ( $x \in \mathbb{R}^d$ ), then this leads to the NW kernel smoother (2.7).

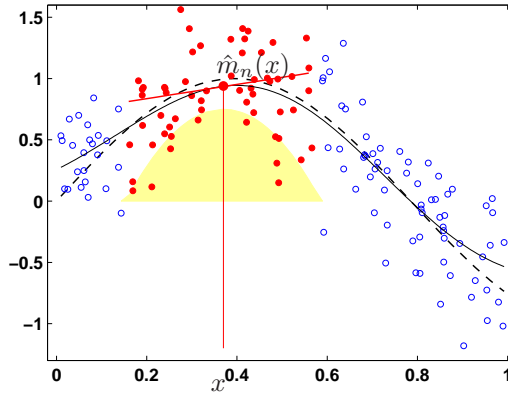
One of the most popular local modeling estimate is the *local polynomial kernel estimate*. Here one fits a polynomial of certain order to the data given by

$$g(x, \{a_k\}_{k=1}^l) = \sum_{k=1}^l a_k x^{k-1}. \quad (2.9)$$

This is a polynomial of degree  $l - 1$  (or less) in  $x$ . Within the framework of the local polynomial kernel estimate often  $l = 2$  is chosen i.e. the *local linear kernel estimate*. This kernel estimate in particular has several advantages over the NW kernel estimate: (i) The method adapts to various types of designs such as random and fixed designs, highly clustered and nearly uniform designs. This is referred to as the *design adaption property*. This is in contrast with the Gasser-Müller estimator which cannot adapt to random design: the unconditional variance is 1.5 times higher for random design. More explanation about this statement can be found in Mack and Müller (1989) and Chu and Marron (1991a); (ii) There is an absence of boundary effects i.e. the bias at the boundary stays automatically of the same order as in the interior without the use of specially designed boundary kernels which is referred to as *automatic boundary carpentry* (Fan and Gijbels, 1992; Ruppert and Wand, 1994). (iii) Also, local polynomial estimators have nice minimax efficiency properties: the asymptotic minimax efficiency for commonly

used orders is 100% among all linear estimators and only a small loss has to be tolerated beyond this class. More theoretical aspects as well as applications and extensions to the multivariate case regarding this class of modeling can be found in Fan and Gijbels (1996) and Loader (1999).

For local linear kernel estimate, it is clear that it corresponds to locally approximating the regression function with a linear model where the weight of the  $Y_i$  depends on the distance between  $X_i$  and  $x$ . Figure 2.3 illustrates the principle of the local linear kernel estimate.



**Figure 2.3:** 100 pairs  $(X_i, Y_i)$  are generated at random from  $Y = \sin(4X)$  (dashed line) with Gaussian errors  $\varepsilon \sim \mathcal{N}(0, 1/3)$  and  $X \sim \mathcal{U}[0, 1]$ . The dot around 0.38 (vertical line) is the fitted constant  $\hat{m}_n(x)$ , and the full circles indicate those observations contributing to the fit at  $x$ . The solid region indicates the weights assigned to observations according to the Epanechnikov kernel. The full local linear kernel estimate is shown by the full line.

The local polynomial estimate has no global consistency properties. Similar to the linear partitioning estimate (Györfi et al., 2002), the local linear kernel estimate  $\hat{m}_n$  is not weakly universally consistent. This holds because due to interpolation effects, the local linear kernel estimate can take arbitrary large values even for bounded data. The counterexample for the consistency of the local polynomial kernel estimate is due to Devroye (personal communication to L. Györfi, 1998). However, this problem can be easily avoided by minimizing (2.8) only over coefficients which are bounded in absolute value by some constant depending on  $n$  and converging to infinity (Kohler, 2002).



### 2.4.4 Global Modeling

Least squares estimates are defined by minimizing the empirical  $L_2$  risk functional over a general set of functions  $\mathcal{F}_n$ . This leads to a function which interpolates the data and hence is not a reasonable estimate. Therefore, one has to restrict the set of functions over which one minimizes the empirical  $L_2$  risk functional. The global modeling estimate is defined as

$$\hat{m}_n(\cdot) = \arg \min_{f \in \mathcal{F}_n} \frac{1}{n} \sum_{i=1}^n |f(X_i) - Y_i|^2,$$

and hence it minimizes the empirical  $L_2$  risk

$$\frac{1}{n} \sum_{i=1}^n |f(X_i) - Y_i|^2.$$

As an examples of this class consider neural networks (Bishop, 1995; Dreyfus, 2005). Given a training data set  $\mathcal{D}_n = \{(X_1, Y_1), \dots, (X_n, Y_n)\}$  and let the parameters of the network chosen to minimize the empirical  $L_2$  risk functional for the class of neural networks

$$\mathcal{F}_n = \left\{ \sum_{i=1}^h \beta_i g(w_i^T X + b_i) + \beta_0 : h \in \mathbb{N}, w_i \in \mathbb{R}^d, b_i \in \mathbb{R}, \sum_{i=1}^h |\beta_i| \leq a_n \right\}, \quad (2.10)$$

where  $g : \mathbb{R} \rightarrow [0, 1]$  is e.g. a sigmoidal or hyperbolic tangent function,  $w_1, \dots, w_h \in \mathbb{R}^d$ ,  $b_1, \dots, b_h \in \mathbb{R}$ ,  $\beta_0, \beta_h \in \mathbb{R}$  are the parameters that specify the network. Notice that the range of some parameters in (2.10) is restricted. This restriction is needed to obtain consistency (Lugosi and Zeger, 1995). The estimate  $\hat{m}_n \in \mathcal{F}_n$  satisfies

$$\frac{1}{n} \sum_{i=1}^n |\hat{m}_n(X_i) - Y_i|^2 = \min_{f \in \mathcal{F}_n} \frac{1}{n} \sum_{i=1}^n |f(X_i) - Y_i|^2$$

Other examples of this class are regression splines (Eubank, 1999; Hastie et al., 2009) and partitioning estimates (Györfi et al., 2002).

### 2.4.5 Penalized Modeling

Instead of restricting the class of functions penalized least squares estimates add a term to the functional to be minimized. This idea, in particular for smoothing splines, goes back to Whittaker (1923), Schoenberg (1964) and Reinsch (1967). This additional term penalizes the *roughness* of a function  $f$ . Let  $v \in \mathbb{N}$ ,  $\lambda_n > 0$  and let the univariate penalized least squares estimate be defined as

$$\hat{m}_n(\cdot) = \arg \min_{f \in C^v(\mathbb{R})} \left\{ \frac{1}{n} \sum_{i=1}^n |f(X_i) - Y_i|^2 + \lambda_n J_{n,v}(f) \right\},$$

where  $C^v(\mathbb{R})$  is the set of all  $v$ -times differentiable functions  $f : \mathbb{R}^d \rightarrow \mathbb{R}$  and  $J_{n,v}(f) = \int |f^{(v)}(u)|^2 du$ . For the penalty term with  $v = 2$ , the minimum is achieved by a cubic spline with knots at the  $X_i$ 's between adjacent values of the  $X_i$ 's (Eubank, 1999; Györfi et al., 2002). This leads to the so-called smoothing spline.

For multivariate  $X$ , the estimate is defined as

$$\hat{m}_n(\cdot) = \arg \min_{f \in W^v(\mathbb{R}^d)} \left\{ \frac{1}{n} \sum_{i=1}^n |f(X_i) - Y_i|^2 + \lambda_n J_{n,v}(f) \right\},$$

where  $W^v(\mathbb{R}^d)$  is the Sobolev space consisting of all functions where weak derivatives of order  $v$  are contained in  $L^2(\mathbb{R}^d)$  (space of square integrable functions over  $\mathbb{R}^d$ ) (Kohler and Krzyżak, 2001) and  $J_{n,v}(f) = \int_{\mathbb{R}^d} \sum_{i_1, \dots, i_k \in \{1, \dots, d\}} \left| \frac{\partial^k f(x)}{\partial x_{i_1} \dots \partial x_{i_k}} \right|^2 dx$  which leads to the so-called thin plate spline estimates.

Further information about spline modeling can be found in Wahba (1990) and references therein.

## 2.5 Support Vector Machines

In this Section we give an overview of Support Vector Machines (SVM) (Vapnik, 1999) and Least Squares Support Vector Machines (LS-SVM) (Suykens et al., 2002). The latter will be the method of choice in this thesis and therefore we discuss this in a separate Section. Roughly speaking, these methods could be classified under the global penalized modeling paradigm. Hastie et al. (2009, Chapter 7) call these type of methods an application of Vapnik's *structural risk minimization program* (Vapnik, 1999). Although the general nonlinear version of SVM is quite recent (Vapnik, 1999, 2000), the roots of the SVM (linear case) approach dates back to the early 1960s (Vapnik and Lerner, 1963; Vapnik and Chervonenkis, 1964) and was originally proposed for classification (separable case). In what follows we will illustrate the idea behind the nonlinear SVM for regression and derive the LS-SVM formulation from it.

### 2.5.1 Basic Idea of Support Vector Machines

The key ingredient of the nonlinear Support Vector Machine (SVM) for classification as well as for regression is the following: let  $\Psi \subseteq \mathbb{R}^{n_f}$  denote a high dimensional (possibly infinite) feature space. Then a random input vector  $X \in \mathbb{R}^d$  is mapped into this high dimensional feature space  $\Psi$  through some mapping  $\varphi : \mathbb{R}^d \rightarrow \Psi$  (in fact there is a relation with the existence of a Hilbert space

$\mathcal{H}$  (Courant and Hilbert, 1953) such that  $\varphi : \mathbb{R}^d \rightarrow \mathcal{H}$  and  $n_f$  is the dimension of  $\mathcal{H}$ . In this space, one considers a class of linear functions defined as

$$\mathcal{F}_{n,\Psi} = \{f : f(X) = w^T \varphi(X) + b, \varphi : \mathbb{R}^d \rightarrow \Psi, w \in \mathbb{R}^{n_f}, b \in \mathbb{R}\}. \quad (2.11)$$

However, even if the linear function in the feature space (2.11) generalizes well, the problem of how to treat the high-dimensional feature space remains. Notice that for constructing the linear function (2.11) in the feature space  $\Psi$ , one does not need to consider the feature space in explicit form i.e. one only needs to replace the inner product in the feature space  $\varphi(X_k)^T \varphi(X_l)$ , for all  $k, l = 1, \dots, n$ , with the corresponding kernel  $K(X_k, X_l)$ . This result is known as Mercer's conditions (Mercer, 1909) and is given in the following theorem.

**Theorem 2.2 (Mercer, 1909)** *Let  $K \in L^2(C)$ ,  $g \in L^2(C)$  where  $C$  is a compact subset of  $\mathbb{R}^d$  and  $K(t, z)$  describes an inner product in some feature space and let  $t, z \in \mathbb{R}^d$ . To guarantee that a continuous symmetric function  $K$  has an expansion*

$$K(t, z) = \sum_{i=1}^{\infty} a_i \phi_i(t) \phi_i(z)$$

with coefficients  $a_i > 0$ . Then it is necessary and sufficient that the condition

$$\iint_C K(t, z) g(t) g(z) dt dz \geq 0$$

is valid for all  $g \in L^2(C)$ .

Using Mercer's condition, one can write  $K(t, z) = \sum_{i=1}^{n_f} \sqrt{a_i} \phi_i(t) \sqrt{a_i} \phi_i(z)$  and define  $\varphi_i(t) = \sqrt{a_i} \phi_i(t)$  and  $\varphi_i(z) = \sqrt{a_i} \phi_i(z)$  such that the kernel function can be expressed as the inner product

$$K(t, z) = \varphi(t)^T \varphi(z). \quad (2.12)$$

Hence, having a positive semidefinite kernel is a condition to guarantee that (2.12) is valid.

## 2.5.2 Primal-Dual Formulation of Support Vector Machines

Given a training data set  $\mathcal{D}_n = \{(X_1, Y_1), \dots, (X_n, Y_n)\}$  and consider the model class  $\mathcal{F}_{n,\Psi}$  defined in (2.11). Following Vapnik (1999), one minimizes the empirical risk functional in the feature space

$$\mathcal{R}_{\text{emp}}(w, b) = \frac{1}{n} \sum_{i=1}^n |Y_i - w^T \varphi(X_i) - b|_{\varepsilon}$$

subject to the constraint  $\|w\| \leq a_n$  with  $a_n \in \mathbb{R}^+$ .  $|\cdot|_\varepsilon$  denotes the Vapnik  $\varepsilon$ -insensitive loss function defined as

$$|Y - f(X)|_\varepsilon = \begin{cases} 0, & \text{if } |Y - f(X)| \leq \varepsilon; \\ |Y - f(X)| - \varepsilon, & \text{otherwise.} \end{cases}$$

The optimization problem (minimization of the empirical risk functional in the feature space)

$$\begin{cases} \min_{w,b} \mathcal{J}_P(w,b) = \frac{1}{n} \sum_{i=1}^n |Y_i - w^T \varphi(X_i) - b|_\varepsilon \\ \text{s.t.} \quad \|w\| \leq a_n \end{cases}$$

is related to the problem of finding  $w$  and  $b$  minimizing the quantity defined by slack variables  $\xi_i, \xi_i^*, i = 1, \dots, n$  and  $c > 0$

$$\boxed{\text{P}} \begin{cases} \min_{w,b,\xi,\xi^*} \mathcal{J}_P(w,\xi,\xi^*) = \frac{1}{2} w^T w + c \sum_{i=1}^n (\xi_i + \xi_i^*) \\ \text{s.t.} \quad Y_i - w^T \varphi(X_i) - b \leq \varepsilon + \xi_i, \quad i = 1, \dots, n, \\ w^T \varphi(X_i) + b - Y_i \leq \varepsilon + \xi_i^*, \quad i = 1, \dots, n, \\ \xi_i, \xi_i^* \geq 0, \quad i = 1, \dots, n. \end{cases} \quad (2.13)$$

The constant  $c > 0$  (notice that  $c$  also depends on  $n$ ) determines the amount up to which deviations from the desired  $\varepsilon$  accuracy are tolerated. This above optimization problem is called an optimization in the *primal* space. Since we do not know  $\varphi(\cdot)$  explicitly, it is impossible to solve (2.13). However, by using the method of Lagrange multipliers (Bertsekas, 1996; Boyd and Vandenberghe, 2004) one only has inner product of the form  $\varphi(X_k)^T \varphi(X_l)$ , for all  $k, l = 1, \dots, n$  which can be replaced by a kernel function  $K(X_k, X_l)$  satisfying Mercer's condition (Theorem 2.2). Hence, after taking the Lagrangian and conditions for optimality, one obtains the following *dual* problem in the dual variables  $\alpha$  and  $\alpha^*$

$$\boxed{\text{D}} \begin{cases} \max_{\alpha, \alpha^*} \mathcal{J}_D(\alpha, \alpha^*) = -\frac{1}{2} \sum_{i,j=1}^n (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) K(X_i, X_j) \\ \quad - \varepsilon \sum_{i=1}^n (\alpha_i - \alpha_i^*) + \sum_{i=1}^n Y_i (\alpha_i - \alpha_i^*) \\ \text{s.t.} \quad \sum_{i=1}^n (\alpha_i - \alpha_i^*) = 0 \\ \quad \alpha_i, \alpha_i^* \in [0, c]. \end{cases} \quad (2.14)$$

The dual representation of the model becomes

$$\hat{m}_n(x) = \sum_{i=1}^n (\alpha_i - \alpha_i^*) K(x, X_i) + b,$$

where  $\alpha_i, \alpha_i^*$  are the solutions to the Quadratic Programming (QP) problem (2.14) and  $b$  follows from the complementary conditions for optimality. The non-zero  $\alpha_i$  are called support vectors. The solution to the QP problem is global and unique provided that the chosen kernel function  $K$  satisfies Mercer's condition (Theorem 2.2). For  $K(X_k, X_l)$  there are usually the following choices:

- linear kernel:  $K(X_k, X_l) = X_k^T X_l$ ,
- polynomial kernel of degree  $k$  with  $c \geq 0$ :  $K(X_k, X_l) = (X_k^T X_l + c)^k$ ,
- Gaussian kernel with bandwidth  $h$ :  $K(X_k, X_l) = (2\pi)^{-d/2} \exp\left(\frac{-\|X_k - X_l\|^2}{2h^2}\right)$ .
- RBF kernel with bandwidth  $h$ :  $K(X_k, X_l) = \exp\left(\frac{-\|X_k - X_l\|^2}{h^2}\right)$ .

There exist numerous methods to solve the QP problem in (2.14) in a fast and numerically stable way e.g. interior point algorithms (Nesterov and Nemirovskii, 1993; Nocedal and Wright, 2006), successive overrelaxation (Mangasarian and Musicant, 1999), sequential minimal optimization (Platt, 1999).

Recently, Christmann and Steinwart (2007) have shown that kernel based regression using a square loss function is weakly universally consistent if  $\mathcal{H}$  is a reproducing kernel Hilbert space of a universal kernel (Micchelli et al., 2006) on  $X$  and  $0 < c_n < \infty$  with  $c_n \rightarrow 0$  and  $c_n^4 n \rightarrow \infty$  for  $n \rightarrow \infty$ . A kernel is called universal if for any prescribed compact subset  $\mathcal{Z}$  of  $\mathcal{X}$ , any positive number  $\varepsilon$  and any function  $f \in C(\mathcal{Z})$  there exist a function  $g \in K(\mathcal{Z})$  such that  $\|f - g\|_{\mathcal{Z}} \leq \varepsilon$ . Here  $\|\cdot\|_{\mathcal{Z}}$  denotes the maximum norm over the compact subset  $\mathcal{Z}$ . Note that universality of kernels can also be expressed in terms of Taylor series and Fourier series (Steinwart, 2001). Theoretical results of Christmann and Steinwart (2007) also show that kernel based regression methods using a loss function with bounded first derivative (e.g. logistic loss) in combination with a bounded and continuous kernel (e.g. Gaussian kernel) are not only consistent and computational tractable, but also offer attractive robustness properties.

Steinwart (2001) showed, in the classification setting, that the soft margin algorithms with universal kernels are consistent for a large class of classification problems including some kind of noisy tasks provided that the regularization parameter is chosen well i.e. independent of the training set size (Steinwart, 2001, Theorem 18, 19 and 24). Finally, Steinwart (2001) shows that even for simple cases, noise free classification problems SVMs with polynomial kernels can behave arbitrarily bad (Steinwart, 2001, Proposition 20). However, in this area some problems are still open i.e. it is interesting whether the soft margin algorithms yield arbitrarily good generalization for all distributions. The results of Steinwart (2001) only provide consistency if the noise level is constant.

### 2.5.3 Least Squares Support Vector Machines

A interesting aspect of SVM is that one solves nonlinear regression (and classification) problems by means of convex quadratic programs. Moreover, one also obtains sparseness as a result of this QP problem. Is it possible to simplify the SVM formulation without loosing any of its advantages? As we will show, this turns out to be true at the loss of sparseness. The following SVM modification was proposed by Suykens et al. (2002): replace the inequality constraints by equality constraints and use a squared loss function instead of an  $\varepsilon$ -insensitive loss function. Given a training data set  $\mathcal{D}_n = \{(X_1, Y_1), \dots, (X_n, Y_n)\}$  where  $X \in \mathbb{R}^d$  and  $Y \in \mathbb{R}$  and consider the model class  $\mathcal{F}_{n, \Psi}$  defined in (2.11) and let  $\gamma > 0$ . Minimizing the empirical risk functional in the feature space with a squared loss leads to the following primal optimization problem

$$\boxed{\text{P}} \begin{cases} \min_{w, b, e} \mathcal{J}_P(w, e) = \frac{1}{2} w^T w + \frac{\gamma}{2} \sum_{i=1}^n e_i^2 \\ \text{s.t.} & Y_i = w^T \varphi(X_k) + b + e_i, \quad i = 1, \dots, n, \end{cases}$$

Note that this cost function consists of a Residual Sum of Squares (RSS) fitting error and a regularization term, which is also a standard procedure for the training of Multi-Layer Perceptrons (MLPs). Also, the above formulation is related to ridge regression (Hoerl and Kennard, 1970; Golub and Van Loan, 1996). The relative importance of these terms is determined by the positive real constant  $\gamma$ . In the case of noisy data one avoids overfitting by taking a smaller  $\gamma$  value.

To solve the optimization problem in the dual space, one defines the Lagrangian

$$\mathcal{L}(w, b, e; \alpha) = \frac{1}{2} w^T w + \frac{\gamma}{2} \sum_{i=1}^n e_i^2 - \sum_{i=1}^n \alpha_i \{w^T \varphi(X_i) + b + e_i - Y_i\},$$

with Lagrange multipliers  $\alpha_i \in \mathbb{R}$  (called support vectors). The conditions for optimality are given by

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial w} = 0 & \rightarrow w = \sum_{i=1}^n \alpha_i \varphi(X_i) \\ \frac{\partial \mathcal{L}}{\partial b} = 0 & \rightarrow \sum_{i=1}^n \alpha_i = 0 \\ \frac{\partial \mathcal{L}}{\partial e_i} = 0 & \rightarrow \alpha_i = \gamma e_i, \quad i = 1, \dots, n \\ \frac{\partial \mathcal{L}}{\partial \alpha_i} = 0 & \rightarrow w^T \varphi(X_i) + b + e_i - Y_i = 0, \quad i = 1, \dots, n. \end{cases}$$

After elimination of  $w$  and  $e$  the solution yields

$$\boxed{\text{D}} \left[ \begin{array}{c|c} 0 & 1_n^T \\ \hline 1_n & \Omega + \frac{1}{\gamma} I_n \end{array} \right] \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ Y \end{bmatrix}, \quad (2.15)$$

with  $Y = (Y_1, \dots, Y_n)^T$ ,  $1_n = (1, \dots, 1)^T$  and  $\alpha = (\alpha_1, \dots, \alpha_n)^T$  the Lagrange multipliers. By using Mercer's condition, the  $kl$ -th element of  $\Omega$  is given by

$$\Omega_{kl} = \varphi(X_k)^T \varphi(X_l) = K(X_k, X_l) \quad k, l = 1, \dots, n,$$

where  $\Omega$  is a positive definite matrix and the  $kl$ -th element of the matrix  $\Omega_{kl} = K(X_k, X_l)$  is a symmetric, continuous function. The kernel function  $K$  can be chosen as  $K(u) = (2\pi)^{-d/2} \exp(-\|u\|^2/2)$  where  $u = (X_k - X_l)/h$  and bandwidth  $h$ . The resulting LS-SVM model is given by

$$\hat{m}_n(x) = \sum_{i=1}^n \alpha_i K(x, X_i) + b. \quad (2.16)$$

As with SVM, one has a global and unique solution. The dual problem for nonlinear LS-SVMs corresponds to solving a linear system which is easier to solve compared to a QP problem with SVM. Unfortunately, a drawback of this simplification is the loss of sparseness. Therefore, in the LS-SVM case, every data point is a support vector. This is immediately clear from the condition for optimality

$$\alpha_i = \gamma e_i, \quad i = 1, \dots, n.$$

However, there are numerous works addressing the sparseness of LS-SVM models based on different approaches. They can be broadly divided into two categories: (i) Pruning after training and then retraining (Suykens et al., 2000; Li et al., 2006). A common problem in both works is that it is not guaranteed that the number of support vectors will be greatly reduced; (ii) Enforcing sparseness from the beginning. Recently, in this last category, López et al. (2011) suggested to use the  $L_0$  norm in order to obtain sparse LS-SVM models via an iterative reweighting scheme since the problem is non-convex.

## 2.6 Conclusions

In this Chapter, we have reviewed the basic properties of parametric and nonparametric modeling. Several model classes were briefly discussed such as local averaging, local modeling, global modeling and penalized modeling. We have described the assumptions and restrictions on the regression estimates and also we have clarified that any estimate can have an arbitrary slow rate of convergence. Further, we have illustrated the curse of dimensionality by means of an example and how it can effect the quality of estimation. Finally, we motivated the basic principle of support vector machines and least squares support vector machines. The latter method will be the method of choice in this thesis.





## Chapter 3

# Model Selection Methods

From Chapter 2 we know that most learning algorithms such as local polynomial regression, SVM, LS-SVM, etc. have some additional tuning parameters. These are often the bandwidth of the kernel and the regularization parameter. In this Chapter we describe and motivate some methods for selecting these additional parameters. As we will show, this is not always an easy task since the designed cost functions for finding these parameters (model selection) are not necessarily convex. In order to select suitable parameters we proposed a methodology (De Brabanter et al., 2010a) consisting of two steps which gives rise to fully automated model selection procedures.

### 3.1 Introduction

Most learning algorithms such as neural networks, local polynomial regression, smoothing splines, SVM, LS-SVM, etc. require the tuning of some extra parameter or parameters. Mostly, in kernel methods these are the bandwidth of the kernel and a regularization parameter. The latter is not always needed e.g. in local polynomial regression, Nadaraya-Watson, . . .

Tuning parameter selection methods can be divided into three classes: (i) Cross-validation (Clark, 1975; Wahba and Wold, 1975; Burman, 1989) and bootstrap (Davison and Hinkley, 2003); (ii) Plug-in methods (Härdle, 1989; Fan and Gijbels, 1996); (iii) Complexity criteria (Mallows, 1973; Akaike, 1973; Schwartz, 1979; Vapnik, 1999). The reason why these parameters have to be determined in such a way is due to the bias–variance tradeoff. Simply, this can be interpreted as follows: With too much fitting, the model adapts itself too closely to the training data and will not generalize well on new or unseen data. In this

case we will have a large variance and small bias. On the other hand, when the model is not complex enough, it will underfit the data and will have a large bias (and small variance) and hence resulting in poor generalization. We will now study this tradeoff in some more detail.

Let  $\hat{m}_n$  be an arbitrary estimate. For any  $x \in \mathbb{R}^d$  the expected squared error of  $\hat{m}_n$  at  $x$  can be written as

$$\begin{aligned} \mathbf{E}[|\hat{m}_n(x) - m(x)|^2] &= \mathbf{E}[|\hat{m}_n(x) - \mathbf{E}[\hat{m}_n(x)]|^2] + |\mathbf{E}[\hat{m}_n(x)] - m(x)|^2 \\ &= \mathbf{Var}[\hat{m}_n(x)] + |\text{bias}[\hat{m}_n(x)]|^2, \end{aligned}$$

where  $\text{bias}[\hat{m}_n(x)]$  is the difference between the expectation of  $\hat{m}_n(x)$  and  $m(x)$ . Similarly, a decomposition of the expected  $L_2$  error yields

$$\mathbf{E} \left[ \int |\hat{m}_n(x) - m(x)|^2 dF(x) \right] = \int \mathbf{Var}[\hat{m}_n(x)] dF(x) + \int |\text{bias}[\hat{m}_n(x)]|^2 dF(x).$$

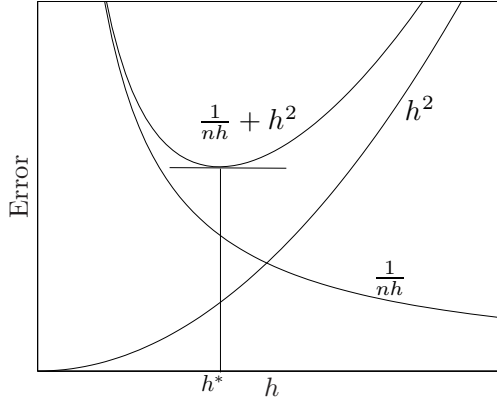
The importance of these decompositions is that the integrated variance and the integrated squared bias depend in opposite ways on the wiggleness of an estimate. If one increases the wiggleness of an estimate, then usually the integrated bias will decrease, but the integrated variance will increase. Under certain regularity conditions (Fan and Gijbels, 1996; Simonoff, 1996) it can be shown for local polynomial regression (with a polynomial of even degree ( $l$  in the sense of (2.9)) that

$$\int_{\mathbb{R}^d} \mathbf{Var}[\hat{m}_n(x)] dF(x) = \frac{\kappa_1}{nh^d} + o\left(\frac{1}{nh^d}\right)$$

and

$$\int_{\mathbb{R}^d} |\text{bias}[\hat{m}_n(x)]|^2 dF(x) = \kappa_2 h^l + o(h^{2l}),$$

where  $h$  denotes the bandwidth of the kernel estimate,  $\kappa_1$  is a constant depending on the conditional variance  $\mathbf{Var}[Y|X = x]$ , the kernel and the design density and  $\kappa_2$  is a constant depending on the kernel and the  $l^{\text{th}}$  derivative of the true regression function. Note that the notation for the odd degree of the polynomial in Fan and Gijbels (1996) equals  $l - 1$  in our notation. Figure 3.1 shows the integrated variance, squared bias and the expected  $L_2$  error for local polynomial regression with  $l = 2$ . The optimal value  $h^*$  for which the sum of the integrated variance and the squared bias is minimal depends on  $\kappa_1$  and  $\kappa_2$ . Since  $\kappa_1$  and  $\kappa_2$  depend on the underlying distribution, it is important to have methods which choose the bandwidth automatically using only the data  $\mathcal{D}_n$ . In the remaining of the Chapter such methods will be discussed.



**Figure 3.1:** Bias-variance tradeoff. The integrated variance, squared bias and the expected  $L_2$  error for local polynomial regression with degree  $l = 2$ .  $h^*$  is the optimal value for which the sum of the integrated variance and the squared bias is minimal.

## 3.2 Cross-Validation Procedures

The purpose of this Section is to describe the rationale behind cross-validation (CV) and to define several CV procedures. For a historical as well as a theoretical overview of CV procedures see Arlot and Celisse (2010).

### 3.2.1 Cross-Validation Philosophy

Larson (1931) already noticed that training an algorithm and evaluating its statistical performance on the same data yields an overoptimistic result. To avoid such results CV procedures were developed (Stone, 1974; Geisser, 1975).

Since in most real life applications the number of data is limited, a technique called splitting the data has to be employed. Part of the data is used for training the algorithm (training data) and the other part is used for evaluating the performance of the model (validation data). Here, the validation data plays the role of new data as long as the data are i.i.d. Various splitting strategies exist e.g. hold-out (Devroye and Wagner, 1979; Devroye et al., 1996), leave-one-out (Stone, 1974; Allen, 1974), leave- $p$ -out (Shao, 1993),  $v$ -fold (Geisser, 1975; Burman, 1989), Monte-Carlo (Picard and Cook, 1984; Marron, 1992), repeated learning testing (Breiman et al., 1984; Burman, 1989).

The major interest of CV lies in the universality of the data splitting heuristics.

It only assumes that data are identically distributed, and training and validation samples are independent which can even be relaxed. Therefore, CV can be applied to many algorithms in (almost) any framework, such as regression (Stone, 1974), density estimation (Rudemo, 1982; Bowman, 1984), and classification (Devroye and Wagner, 1979).

### 3.2.2 Leave-One-Out Cross-Validation

Consider the integrated squared error (ISE) as a measure of accuracy for the estimator  $\hat{m}_n(x; \theta)$ , let  $\theta$  denote the tuning parameter(s) of the estimator e.g.  $\theta = h$  in the NW kernel smoother or  $\theta = (h, \gamma)$  for LS-SVM, and given a data set  $\mathcal{D}_n = \{(X_1, Y_1), \dots, (X_n, Y_n)\}$ . Note that all elements of  $\theta$  should be strictly positive. The main idea is to construct estimates  $\hat{m}_n(x; \theta)$  such that the ISE is small. Let  $F$  denote the distribution over the input space, then

$$\begin{aligned} \int |\hat{m}_n(x; \theta) - m(x)|^2 dF(x) &= \int m^2(x) dF(x) + \int \hat{m}_n^2(x; \theta) dF(x) \\ &\quad - 2 \int \hat{m}_n(x; \theta) m(x) dF(x). \end{aligned} \quad (3.1)$$

Since the first term in (3.1) is independent of  $\theta$ , minimizing (3.1) is equivalent to minimizing

$$\int \hat{m}_n^2(x; \theta) dF(x) - 2 \int \hat{m}_n(x; \theta) m(x) dF(x). \quad (3.2)$$

In practice this would be impossible to compute since this quantity depends on the unknown real-valued (true) function  $m$  and the density  $f$ . The first term of (3.2) can be entirely computed from the data  $\mathcal{D}_n$  and the second term can be written as

$$\int \hat{m}_n(x; \theta) m(x) dF(x) = \mathbf{E}[\hat{m}_n(x; \theta) m(x) | \mathcal{D}_n]. \quad (3.3)$$

If one estimates (3.3) by its empirical version  $n^{-1} \sum_{i=1}^n Y_i \hat{m}_n(X_i; \theta)$  the selection will be a biased estimator of the ISE. The bias is due to the fact that the observation  $Y_i$  is used in  $\hat{m}_n(X_i; \theta)$  to predict itself. However, there exist several methods to find an unbiased estimate of the ISE e.g. plug-in methods, leave-one-out (LOO) technique and a modification so that bias cancels out asymptotically. Here we will use the LOO technique in which one observation is left out. Therefore, a better estimator for (3.3) instead of its straight empirical version is

$$\frac{1}{n} \sum_{i=1}^n Y_i \hat{m}_n^{(-i)}(X_i; \theta), \quad (3.4)$$

where  $\hat{m}_n^{(-i)}(X_i; \theta)$  denotes the LOO estimator with point  $i$  left out from the training. Similarly, the first term of (3.2) can be written as

$$\frac{1}{n} \sum_{i=1}^n \left| \hat{m}_n^{(-i)}(X_i; \theta) \right|^2. \quad (3.5)$$

From (3.4) and (3.5), the LOO-CV function is given by

$$\text{CV}(\theta) = \frac{1}{n} \sum_{i=1}^n \left| Y_i - \hat{m}_n^{(-i)}(X_i; \theta) \right|^2.$$

The LOO cross-validated selection of  $\theta$  is

$$\hat{\theta} = \arg \min_{\theta} \frac{1}{n} \sum_{i=1}^n \left| Y_i - \hat{m}_n^{(-i)}(X_i; \theta) \right|^2.$$

### 3.2.3 v-fold Cross-Validation

In general there is no reason that training sets should be of size  $n - 1$  as in the LOO-CV case. There is the possibility that small perturbations, when single observations are left out, make  $\text{CV}(\theta)$  too variable, if the fitted values  $\hat{m}_n(x; \theta)$  do not depend smoothly on the empirical distribution  $\hat{F}_n$  or if the loss function  $L(Y, \hat{m}_n(X; \theta))$  is not continuous. These potential problems can be avoided to a large extent by leaving out groups of observations, rather than single observations. Also, it offers a computational advantage since we do not have to compute  $n$  estimates but only  $v$  in  $v$ -fold CV. The latter plays an important role for large data sets.

For  $v$ -fold CV the splits are chosen in a special deterministic way. Let  $1 \leq v \leq n$  and assume that  $n/v$  is an integer. Divide the data  $\mathcal{D}_n$  into  $v$  disjoint groups of equal size  $n/v$  and denote the set of data consisting of all groups, except the  $l^{\text{th}}$  one, by  $\mathcal{D}_{n,l}$ :

$$\mathcal{D}_{n,l} = \left\{ (X_1, Y_1), \dots, (X_{\frac{n}{v}(l-1)}, Y_{\frac{n}{v}(l-1)}), (X_{\frac{n}{v}l+1}, Y_{\frac{n}{v}l+1}), \dots, (X_n, Y_n) \right\}.$$

For each data set  $\mathcal{D}_{n,l}$  construct estimates  $\hat{m}_n^{(-\mathcal{D}_{n,l})}(\cdot; \theta)$ . Choose  $\theta$  such that

$$\frac{1}{v} \sum_{l=1}^v \frac{1}{n/v} \sum_{i=\frac{n}{v}(l-1)+1}^{\frac{n}{v}l} \left| Y_i - \hat{m}_n^{(-\mathcal{D}_{n,l})}(X_i; \theta) \right|^2.$$

is minimal. Also note that  $v$ -fold CV with  $v = n$  is LOO-CV.

The use of groups have the desired effect of reducing variance, but at the cost of increasing bias. According to Beran (1984) and Burman (1989) the bias of  $v$ -fold CV yields

$$a_0 [(v-1)^{-1}n^{-1}].$$

For LOO-CV the bias is of order  $O(n^{-2})$ , but when  $v$  is small the bias term is not necessarily small. Therefore, the use of 2-fold CV is never recommended. The term  $a_0$ , depending on the loss function  $L$  used in the CV procedure and the empirical distribution  $\hat{F}_n$ , is of the order of the number of effective parameters being estimated. As a result, if the number of effective parameters is not small, the  $v$ -fold CV is a poor estimate of the prediction error. However, there are adjustments possible to reduce the bias in  $v$ -fold CV, see e.g. Burman (1989, 1990); Tibshirani and Tibshirani (2009); Arlot and Celisse (2010). These adjustments to the  $v$ -fold CV procedure reduce the bias to

$$a_1 [(v-1)^{-1}n^{-2}],$$

for some constant  $a_1$  depending on the loss function  $L$  used in the CV procedure and the empirical distribution  $\hat{F}_n$ .

Precise understanding of how  $\mathbf{Var}[v\text{-fold CV}]$  depends on the splitting scheme is rather complex since the number of splits (folds)  $v$  is linked with the number of points used as validation. Furthermore, the variance of CV strongly depends on the framework and on the stability of the algorithm. Therefore, radically different results have been obtained in different frameworks, in particular on the value of  $v$  for which the  $v$ -fold CV estimator has a minimal variance, see e.g. Burman (1989) and Hastie et al. (2009, Chapter 7).

What is a suitable value for  $v$ ? Davison and Hinkley (2003) have suggested the following rule of thumb. Take  $v = \min(\sqrt{n}, 10)$ , because taking  $v > 10$  maybe computationally too expensive while taking groups of size at least  $\sqrt{n}$  should perturb the data sufficiently to give a small variance of the estimate.

### 3.2.4 Generalized Cross-Validation

Generalized Cross-Validation (GCV) was first proposed by Craven and Wahba (1979) in the context of nonparametric regression with a roughness penalty. However, Golub et al. (1979) showed that GCV can be used to solve a wide variety of problems. Before formulating the GCV estimator we need the following definition first.

**Definition 3.1 (Linear smoother)** *An estimator  $\hat{m}_n$  of  $m$  is called a linear smoother if, for each  $x \in \mathbb{R}^d$ , there exists a vector  $L(x) = (l_1(x), \dots, l_n(x))^T \in \mathbb{R}^n$*

such that

$$\hat{m}_n(x) = \sum_{i=1}^n l_i(x) Y_i.$$

In matrix form, this can be written as  $\hat{m}_n = LY$ , with  $L \in \mathbb{R}^{n \times n}$  and  $L$  is called the smoother matrix. Craven and Wahba (1979) showed that the deleted residuals  $Y_i - \hat{m}_n^{(-i)}(X_i; \theta)$  can be written in terms of  $Y_i - \hat{m}_n(X_i; \theta)$  and the trace of the smoother matrix  $L$ . Also note that the smoother matrix depends on  $\theta$ . The GCV criterion satisfies

$$GCV(\theta) = \frac{1}{n} \sum_{i=1}^n \left| \frac{Y_i - \hat{m}_n(X_i; \theta)}{1 - n^{-1} \text{tr}[L(\theta)]} \right|^2. \tag{3.6}$$

The GCV estimate of  $\theta$  can be found by minimizing (3.6).

GCV is actually closer to  $C_L$ , i.e.  $C_p$  generalized for linear estimators (Mallows, 1973) than CV, since GCV can be seen as an approximation to  $C_p$  with a particular estimator of variance (Efron, 1986). The efficiency of GCV has been investigate in Li (1987) and Cao and Golubev (2006). An interesting fact regarding GCV is that for a given data set, GCV always selects the same  $\theta$ , no matter the magnitude of noise.

### 3.3 Complexity Criteria: Final Prediction Error, AIC, Mallows' $C_p$ and BIC

Let  $\mathcal{P}$  be a finite set of parameters. For  $\theta \in \mathcal{P}$ , let  $\mathcal{F}_{n,\theta}$  be a set of functions

$$\mathcal{F}_{n,\theta} = \left\{ m : m(x,\theta) = \theta_0 + \sum_{l=1}^d \theta_l x^l, x \in \mathbb{R}^d \text{ and } \theta \in \mathcal{P} \right\},$$

let  $Q_n(\theta) \in \mathbb{R}^+$  be a complexity term for  $\mathcal{F}_{n,\theta}$  and let  $\hat{m}_n$  be an estimator of  $m$  in  $\mathcal{F}_\theta$ . The learning parameters are chosen to be the minimizer of a cost function defined as

$$J(\theta) = \frac{1}{n} \sum_{k=1}^n \mathcal{L}(y_k, \hat{m}_n(x_k; \theta)) + \lambda(Q_n(\theta)) \hat{\sigma}^2$$

where  $\mathcal{L}$  denotes the loss function,  $Q_n(\theta) \in \mathbb{R}^+$  is a complexity term,  $\lambda > 0$  is a cost complexity parameter and the term  $\hat{\sigma}^2$  is an estimate of the error variance. The Final Prediction Error (FPE) criterion depends only on  $\hat{m}_n$  and the data. If  $\hat{m}_n$  is defined by minimizing the empirical  $L_2$  risk over some linear vector space  $\mathcal{F}_{n,\theta}$  of functions with dimension  $d_\theta$  (number of estimated parameters), then  $J(\theta)$  will be of the form:

- Let  $\lambda = 2$  and  $Q_n(\theta) = n^{-1}d_\theta$ , then Mallows'  $C_p$  is defined as

$$C_p(\theta) = \frac{1}{n} \sum_{k=1}^n |Y_k - \hat{m}_n(X_k; \theta)|^2 + 2 \left( \frac{d_\theta}{n} \right) \hat{\sigma}^2. \quad (3.7)$$

The Akaike Information Criterion (AIC) is similar to (3.7) but more generally applicable when a log-likelihood loss function is used (Hastie et al., 2009). For the Gaussian model (with variance  $\sigma^2 = \hat{\sigma}^2$  assumed known), the AIC statistic is equivalent to  $C_p$ , and is often referred collectively as AIC.

- Let  $\lambda = \log n$  and  $Q_n(\theta) = n^{-1}d_\theta$ , then the Bayesian Information Criterion (BIC) is defined as

$$\text{BIC}(\theta) = \frac{1}{n} \sum_{k=1}^n |Y_k - \hat{m}_n(X_k; \theta)|^2 + \log n \left( \frac{d_\theta}{n} \right) \hat{\sigma}^2.$$

The Bayesian information criterion, like AIC, is applicable in settings where the fitting is carried out by maximization of a log-likelihood. Therefore BIC is proportional to AIC and  $C_p$ , with the factor 2 replaced by  $\log n$ . Assuming  $n > e^2 \approx 7.4$ , BIC tends to penalize complex models more heavily, giving preference to simpler models in selection. Despite its similarity with AIC, BIC is motivated in quite a different way. It arises in the Bayesian approach to model selection (Schwartz, 1979).

- Let  $\lambda = \log(\log n)$  and  $Q_n(\theta) = n^{-1}d_\theta$ , then the Hannan-Quinn Information Criterion (HQIC) (Hannan and Quinn, 1979) is defined as

$$\text{HQIC}(\theta) = \frac{1}{n} \sum_{k=1}^n |Y_k - \hat{m}_n(X_k; \theta)|^2 + \log(\log n) \left( \frac{d_\theta}{n} \right) \hat{\sigma}^2.$$

The HQIC was proposed in the context of autoregressive model order determination.

- Some other complexity criteria are Takeuchi's Information Criterion (TIC) (Takeuchi, 1976), Minimum Description Length (MDL) (Rissanen, 1983) and Vapnik-Chervonenkis (VC) dimension (Vapnik, 2000).

In case of linear regression and time series models, Hurvich and Tsai (1989) showed for small samples that the bias of AIC can be quite large especially as the dimension of the candidate model approaches the sample size (leading to overfitting of the model). In order to overcome this drawback they proposed a corrected AIC (AICC), which was found to be less biased than the classical AIC. The AICC is given by

$$\text{AICC}(\theta) = \frac{1}{n} \sum_{k=1}^n |Y_k - \hat{m}_n(X_k; \theta)|^2 + \left( 1 + \frac{2(d_\theta + 1)}{n - d_\theta - 2} \right) \hat{\sigma}^2. \quad (3.8)$$



The AIC was originally proposed for parametric models as an approximately unbiased estimate of the expected Kullback-Leibler (KL) divergence. Extensions to the nonparametric case are also possible. In this case let  $\mathcal{Q}$  denote a finite set of parameters and let  $\theta \in \mathcal{Q}$ . Let  $\mathcal{F}_{n,\theta}$  be a set of functions satisfying

$$\mathcal{F}_{n,\theta} = \{m : m(X,\theta), X \in \mathbb{R}^d, Y \in \mathbb{R}, \theta \in \mathcal{Q} \text{ and } \hat{m}_n = L(\theta)Y\},$$

where  $L$  denotes the smoother matrix (depending on  $\theta$ ), see (3.1), and  $\hat{m}_n$  is an estimator of  $m$  in  $\mathcal{F}_{n,\theta}$ . The above set of functions are valid for splines, wavelets, local polynomial regression, LS-SVM etc. and are called linear estimators or linear smoothers in the sense that  $\hat{m}_n = L(\theta)Y$  where the smoother matrix  $L(\theta)$  depends on  $X \in \mathcal{D}_n$  and not on  $Y$ . Analogously to the formulation of the AICC for parametric models and based on the effective numbers of parameters (also called effective degrees-of-freedom) used in the nonparametric fit (Hastie and Tibshirani, 1990; Moody, 1992; Hastie et al., 2009), the tuning parameters are chosen to be the minimizer of

$$\text{AICC}(\theta) = \frac{1}{n} \sum_{k=1}^n |Y_k - \hat{m}_n(X_k; \theta)|^2 + \left(1 + \frac{2(\text{tr}[L(\theta)] + 1)}{n - \text{tr}[L(\theta)] - 2}\right) \hat{\sigma}^2,$$

where it is assumed that  $\text{tr}[L(\theta)] < n - 2$ .

## 3.4 Choosing the Learning Parameters

### 3.4.1 General Remarks

In most practical cases it is often preferable to have a data-driven method to select learning parameters. For this selection process, many data-driven procedures have been discussed in the literature. Commonly used are those based on the cross-validation criterion (leave-one-out and  $v$ -fold) and the generalized cross-validation criterion. One advantage of cross-validation and generalized cross-validation over some other selection criteria such as Mallows'  $C_p$ , AIC, AICC, HQIC, TIC and BIC is that they do not require estimates of the error variance. As a consequence, these selection criteria require a roughly correct working model to obtain the estimate of the error variance while this is not a requirement for CV. The difficulty with leave-one-out cross-validation is that it can become computationally very expensive in practical problems involving large data sets. Especially for this reason  $v$ -fold CV was introduced.

Closely related to CV are the bootstrap bandwidth selectors (Efron, 1982). In fact, the bootstrap procedures are nothing more than smoothed versions of cross-validation, with some adjustments made to correct for bias. The improvement of the bootstrap estimators over cross-validation, in the dichotomous situation where

both  $Y$  and the prediction rule are either 0 or 1, is due mainly to the effect of smoothing. In smoother prediction problems, when  $Y$  and the prediction rule are continuous, there is little difference between cross-validation and bootstrap methods. Finally, the choice which criterion to use will depend on the situation.

It is well known that data-driven regression smoothing parameters (bandwidth)  $\hat{\theta}$  based on CV methods and related methods exhibit a slow rate of convergence to their optimum. This rate can be as slow as  $n^{-1/10}$  (Härdle et al., 1988) i.e. for a bandwidth  $\hat{\theta}_0$  optimizing the averaged squared error,  $n^{1/10}(\hat{\theta} - \hat{\theta}_0)/\hat{\theta}_0$  tends to an asymptotic normal distribution. In order to improve this rate Härdle et al. (1992) consider the mean averaged squared error optimal bandwidths  $h_0$ . This nonrandom smoothing parameter can be approximated much faster. They used double smoothing to show that there is a  $\hat{\theta}$  such that, under certain conditions (compactly supported kernels of orders  $r$  and  $s$ ,  $m^{(r+\max(r,s))}$  is continuous on  $(0,1)$  and  $\hat{\sigma}^2$  is  $\sqrt{n}$  consistent for  $\sigma^2$ ),  $n^{1/2}(\hat{\theta} - \theta_0)/\theta_0$  tends to an asymptotic normal distribution.

Data-driven methods such as CV can be computationally quite expensive when considering large data sets. Numerous attempts have been reported in literature in order to reduce the computational burden. However, these speed-up techniques are designed for a specific nonparametric regression smoother. In the case of LS-SVM, consider the works of Cawley and Talbot (2004) and Ying and Keong (2004) for fast LOO-CV and An et al. (2007) for fast  $v$ -fold CV.

Another approach to smoothing introduced by Chaudhuri and Marron (1999) and Chaudhuri and Marron (2000) is called scale-space smoothing that avoids the idea of selecting a single bandwidth. But rather than choosing a single bandwidth, they examine  $\hat{m}_n$  over a set of bandwidths  $\theta$  as a way of exploring the scale-space surface

$$\mathcal{S} = \{m_n(x), x \in \mathcal{X}, \theta \in \mathcal{P}\}.$$

One way to summarize the estimated scale-space surface

$$\hat{\mathcal{S}} = \{\hat{m}_n(x), x \in \mathcal{X}, \theta \in \mathcal{P}\}$$

is to isolate important shape summaries. Chaudhuri and Marron (1999) look for points  $x$  where  $m'_n(x) = 0$  by using  $\hat{m}'_n(x)$  as a set of test statistics. They call the resulting method SiZer (significant zero crossings of derivatives). The method is also a very useful tool for inferring significant features as opposed to being spurious sampling artifacts.

### 3.4.2 Optimization Strategy

In the previous Sections we have discussed several methods that enable and assist the user to choose the tuning parameters. This is a rather difficult task

in practice since the cost functions, described in the previous Section, are non-smooth and therefore can contain multiple local minima. The theoretical results of Hall and Marron (1991) provide an explanation and quantification of this empirical observation through modeling the cross-validation function as a Gaussian stochastic process in the context of density estimation. They show asymptotically that the degree of wiggleness of the cross-validation function depends on the underlying density, but conclude that the dependence of the kernel function is much more complicated.

A typical method to estimate the tuning parameters (finding the minimum value of the CV cost function) would define a grid (grid-search) over these parameters of interest and perform CV for each of these grid values. However, three disadvantages come up with this approach (Bennett et al., 2006; Kunapuli et al., 2008). A first disadvantage of such a grid-search CV approach is the limitation of the desirable number of tuning parameters in a model, due to the combinatorial explosion of grid points. A second disadvantage of this approach is their practical inefficiency, namely, they are incapable of assuring the overall quality of the produced solution. A third disadvantage in grid-search is that the discretization fails to take into account the fact that the tuning parameters are continuous.

In order to overcome these drawbacks we propose a methodology consisting of two steps: first, determine good initial start values by means of a state of the art global optimization technique and second, perform a fine-tuning derivative-free simplex search (Nelder and Mead, 1965; Lagarias et al., 1998) using the previous result as start value. In order to determine good initial starting values, we use the method of Coupled Simulated Annealing with variance control (CSA) (Xavier-de-Souza et al., 2010), see Appendix A for a brief review of the algorithm. Global optimization methods are typically very slow. For many difficult problems, ensuring convergence to a global optimum might mean impractical running times. For such problems, a reasonable solution might be enough in exchange for a faster convergence. Precisely for this reason, many Simulated Annealing (SA) algorithms (Ingber, 1989; Rajasekaran, 2000) and other heuristic based techniques have been developed. However, due to speed-up procedures, these methods often get trapped in poor optima. The CSA method is designed to easily escape from local optima and thus improves the quality of solution without compromising too much the speed of convergence.

The working principle of CSA was inspired by the effect of coupling in Coupled Local Minimizers (CLM) (Suykens et al., 2001) when compared to the uncoupled case i.e. multi-start based methods. CLM and CSA have already proven to be more effective than multi-start gradient descent optimization (Suykens et al., 2001, 2003). Another advantage of CSA is that it uses the acceptance temperature to control the variance of the acceptance probabilities with a control scheme that can be applied to an ensemble of optimizers. This leads to an improved optimization efficiency because it reduces the sensitivity of the algorithm to the

initialization parameters while guiding the optimization process to quasi-optimal runs. This initial result is then used as a starting value for a derivative-free simplex search. This extra step is a fine-tuning procedure resulting in more optimal tuning parameters and hence better performance. This optimization method will be the method of choice in this thesis to find the tuning parameters.

## 3.5 Conclusions

In this Chapter we have given an overview of data-driven model selection methods and complexity criteria. Often in practice the chosen criterion will depend on the situation e.g. small or large data set, can we obtain a good noise variance estimation? We have also illustrated why these methods should be used in order to acquire suitable tuning parameters with respect to the bias-variance tradeoff. Finally, we made clear that minimizing these criteria is often troublesome since there can be multiple local minima present. In order to remove the drawbacks of a grid-search we have proposed a method consisting of two steps giving rise to fully automated model selection procedures.

## Chapter 4

# Fixed-Size Least Squares Support Vector Machines

In this Chapter, we show that solving LS-SVMs for large data sets is computational as well as memory intensive. In order to handle larger data sets, we introduce the so called fixed-size approach where a finite feature map can be approximated by the Nyström method. Suitable selection of the prototype vectors for this approach is performed by maximizing the Rényi entropy where the bandwidth of the kernel is determined via the solve-the-equation plug-in method. Contributions are made in Section 4.3, Section 4.4 and Section 4.5.

### 4.1 Introduction

From Chapter 2 we know that SVM leads to solving a QP problem. Unfortunately, the design of QP solvers e.g. MINOS and LOQO assume that the full kernel matrix is readily available. To overcome this difficulty, decomposition methods (Osuna et al., 1997; Saunders et al., 1998; Joachims, 1999) were designed. A particular case of the decomposition method is iterative chunking where the full scale problem is restricted to a small subset of training examples called the working set. An extreme form of chunking is Sequential Minimal Optimization (SMO) proposed by Platt (1999). SMO uses the smallest possible working set size, i.e. two elements. This choice greatly simplifies the method. Due to this reason SMO is considered as the current state-of-the-art QP solver for solving medium as well as large-scale SVMs.

In the LS-SVM formulation the inequality constraints are replaced by equality constraints and a sum of squared errors cost function is used. Due to the use

of equality constraints and  $L_2$  cost function in LS-SVM, the solution is found by solving a linear system instead of a QP problem. To tackle large-scale problems with LS-SVM Suykens and Vandewalle (1999) and Van Gestel et al. (2004) effectively employed the Hestenes-Stiefel conjugate gradient algorithm (Golub and Van Loan, 1996; Suykens and Vandewalle, 1999). This method is well suited for problems with a larger number of data (up to about 10.000 data points). As an alternative, an iterative algorithm for solving large-scale LS-SVMs was proposed by Keerthi and Shevade (2003). This method is based on the solution of the dual problem using a similar idea to that of the SMO algorithm, i.e. using Wolfe duality theory, for SVMs. The vast majority of textbooks and articles discussing SVMs and LS-SVMs first state the primal optimization problem and then go directly to the dual formulation, see e.g. Vapnik (1999) and Suykens and Vandewalle (1999).

A successful attempt for solving LS-SVMs in primal weight space resulting in a parametric model and sparse representation, introduced by Suykens et al. (2002), is called Fixed-Size Least Squares Support Vector Machines (FS-LSSVM). In this method an explicit expression of the feature map or an approximation to it is required. A procedure to find this approximated feature map is based on the Nyström method (Nyström, 1930; Baker, 1977) which was also used by Williams and Seeger (2001) to speed up Gaussian Processes.

## 4.2 Estimation in the Primal Space

In the following Section we review how to estimate a finite approximation to the feature map  $\varphi$  for LS-SVMs, see Chapter 2, using the Nyström method. The computed finite approximation will then be used to solve the problem in primal space.

### 4.2.1 Finite Approximation to the Feature Map

Recall that the primal optimization problem for LS-SVM was given by

$$\boxed{\text{P}} \left\{ \begin{array}{l} \min_{w,b,e} \mathcal{J}_P(w,e) = \frac{1}{2}w^T w + \frac{\gamma}{2} \sum_{i=1}^n e_i^2 \\ \text{s.t.} \quad Y_i = w^T \varphi(X_k) + b + e_i, \quad i = 1, \dots, n. \end{array} \right. \quad (4.1)$$

Mostly, this problem is solved by constructing the Lagrangian (see Chapter 2). However, if one could obtain an explicit expression for the feature map  $\varphi : \mathbb{R}^d \rightarrow \mathcal{H}$  the above minimization problem can be solved in the primal space. We will show that a finite approximation  $\hat{\varphi} : \mathbb{R}^d \rightarrow \mathbb{R}^m$  of  $\varphi$  is possible and in general  $m \ll n$ .

Let  $X_k \in \mathbb{R}^d$ ,  $k = 1, \dots, n$  be a random sample from an unknown distribution  $F_X$ . Let  $C$  be a compact subset of  $\mathbb{R}^d$ ,  $V = L^2(C)$  and  $M(V, V)$  be a class of linear operators from  $V$  into  $V$ . Consider the eigenfunction expansion of a kernel function

$$K(x, t) = \sum_i \lambda_i \phi_i(x) \phi_i(t),$$

where  $K(x, t) \in V$ ,  $\lambda_i \in \mathbb{C}$  and  $\phi_i \in V$  are respectively the eigenvalues and the eigenfunctions, defined by the Fredholm integral equation of the first kind

$$\begin{aligned} (T\phi_i)(t) &= \int_C K(x, t) \phi_i(x) dF_X(x) \\ &= \lambda_i \phi_i(t), \end{aligned} \quad (4.2)$$

where  $T \in M(V, V)$ .

Then (4.2) can be discretized on a finite set of evaluation points  $\{X_1, \dots, X_n\} \in C \subseteq \mathbb{R}^d$  with associated weights  $v_k \in \mathbb{R}$ ,  $k = 1, \dots, n$ . Define a quadrature method  $Q_n$ ,  $n \in \mathbb{N}$

$$Q_n = \sum_{k=1}^n v_k \psi(X_k).$$

Let  $v_k = \frac{1}{n}$ ,  $k = 1, \dots, n$ , the Nyström method (Nyström, 1930) approximates the integral by means of  $Q_n$  and determines an approximation  $\phi_i$  by

$$\lambda_i \phi_i(t) \approx \frac{1}{n} \sum_{k=1}^n K(X_k, t) \phi_i(X_k), \forall t \in C \subseteq \mathbb{R}^d. \quad (4.3)$$

Let  $t = X_j$ , in matrix notation one obtains

$$\Omega U = U \Lambda,$$

where  $\Omega_{kj} = K(X_k, X_j)$  are the elements of the kernel matrix,  $U = (u_1, \dots, u_n)$  is a  $n \times n$  matrix of eigenvectors of  $\Omega$  and  $\Lambda$  is a  $n \times n$  diagonal matrix of nonnegative eigenvalues in a decreasing order. Expression (4.3) delivers direct approximations of the eigenvalues and eigenfunctions for the  $x_k \in \mathbb{R}^d$ ,  $k = 1, \dots, n$  points

$$\phi_i(x_j) \approx \sqrt{n} u_{i,n} \quad \text{and} \quad \lambda_i \approx \frac{1}{n} \lambda_{i,n}, \quad (4.4)$$

where  $\lambda_{i,n}$  and  $u_{i,n}$  denote the  $i$ th eigenvalue and the  $i$ th eigenvector of (4.3) respectively (the subscript  $n$  denotes the eigenvalues and eigenvectors of (4.3) based on the complete data set).  $\lambda_i$  denote the eigenvalues of (4.2). Substituting (4.4) in (4.3) results in an approximation of an eigenfunction evaluation in point  $t \in C \subseteq \mathbb{R}^d$

$$\hat{\phi}_i(t) \approx \frac{\sqrt{n}}{\lambda_{i,n}} \sum_{k=1}^n K(X_k, t) u_{ki,n},$$

with  $u_{ki,n}$  the  $k^{\text{th}}$  element of the  $i^{\text{th}}$  eigenvector of (4.3). Based on the Nyström approximation, an expression for  $i^{\text{th}}$  entry of the  $n$ -approximated finite feature map  $\hat{\varphi} : \mathbb{R}^d \rightarrow \mathbb{R}^n$  is given by

$$\begin{aligned}\hat{\varphi}_i(x) &= \sqrt{\lambda_i} \hat{\phi}_i(X) \\ &= \frac{1}{\sqrt{\lambda_{i,n}}} \sum_{k=1}^n u_{ki,n} K(X_k, x).\end{aligned}\quad (4.5)$$

This method was used in Williams and Seeger (2001) to speed up Gaussian processes. However, Williams and Seeger (2001) used the Nyström method to approximate the eigenvalues and eigenvectors of the complete kernel matrix by using a random subset of size  $m \ll n$ . A major difference is also that we estimate here in the primal space which leads to a sparse representation.

## 4.2.2 Solving the Problem in Primal Space

In order to introduce parsimony, one can choose a fixed-size  $m$  ( $m \ll n$ ) as a working subsample (see Section 4.3 on how to choose this sample). A likewise  $m$ -approximation to (4.5) can be made yielding

$$\hat{\varphi}_i(x) = \frac{1}{\sqrt{\lambda_{i,m}}} \sum_{k=1}^m u_{ki,m} K(X_k, x).\quad (4.6)$$

Given a data set  $\mathcal{D}_n = \{(X_1, Y_1), \dots, (X_n, Y_n)\}$  and the finite  $m$ -dimensional approximation to the feature map  $\hat{\varphi}$ , (4.1) can be written as

$$\min_{\tilde{w}, b} \frac{1}{2} \sum_{i=1}^m \tilde{w}_i^2 + \gamma \frac{1}{2} \sum_{k=1}^n |Y_k - \tilde{w}^T \hat{\varphi}(X_k) - b|^2,\quad (4.7)$$

with unknowns  $\tilde{w} \in \mathbb{R}^m, b \in \mathbb{R}$  and  $m$  the number of prototype vectors (PVs). Notice that this result is exactly the same in case of classification. The solution to (4.7) is given by

$$\begin{pmatrix} \hat{\tilde{w}} \\ \hat{b} \end{pmatrix} = \left( \hat{\Phi}_e^T \hat{\Phi}_e + \frac{I_{m+1}}{\gamma} \right)^{-1} \hat{\Phi}_e^T Y,\quad (4.8)$$

where  $\hat{\Phi}_e$  is the  $n \times (m+1)$  extended feature matrix

$$\hat{\Phi}_e = \begin{pmatrix} \hat{\varphi}_1(X_1) & \cdots & \hat{\varphi}_m(X_1) & 1 \\ \vdots & \ddots & \vdots & \vdots \\ \hat{\varphi}_1(X_n) & \cdots & \hat{\varphi}_m(X_n) & 1 \end{pmatrix},\quad (4.9)$$

and  $I_{m+1}$  the  $(m+1) \times (m+1)$  identity matrix.



### 4.3 Active Selection of a Subsample

Instead of using a purely random selection of PVs an entropy based selection method is discussed as introduced by Suykens et al. (2002). It is especially important to select the PVs. The selected PVs should represent the main characteristics of the whole training data set, i.e. they take a crucial role in constructing the FS-LSSVM model. Further in this Section it will be illustrated, by means of a toy example, why this criterion is preferred over a purely random selection. This active selection of PVs, based on entropy, refers to finding a subset of size  $m$ , with  $m \ll n$ , of the columns in the kernel matrix that best approximate the kernel matrix.

First, two entropy criteria will be discussed: Shannon (Shannon, 1948; Vollbrecht and Wolf, 2002) and Rényi (Rényi, 1961; Principe et al., 2000; Girolami, 2002; Vollbrecht and Wolf, 2002). See also Beirlant et al. (1997) for a thorough overview concerning nonparametric entropy estimation. Second, we review the solve-the-equation plug-in method introduced by Sheather and Jones (1991), to select the smoothing parameter for entropy estimation. Third, we show how the previous method can be used to determine  $d$  smoothing parameters (bandwidths) if  $X \in \mathbb{R}^d$ . We investigate this for the use in FS-LSSVM for large scale applications.

#### 4.3.1 Subsample Based on Entropy Criteria

Let  $X_k \in \mathbb{R}^d$ ,  $k = 1, \dots, n$  be a set of input samples from a random variable  $X \in \mathbb{R}^d$ . The success of a selection method depends on how much information about the original input sample  $X_k \in \mathbb{R}^d$ ,  $k = 1, \dots, n$ , is contained in a subsample  $X_j \in \mathbb{R}^d$ ,  $j = 1, \dots, m$  ( $m \ll n$ ). In other words, the purpose of a subsample selection is to extract  $m$  ( $m \ll n$ ) samples from  $\{X_1, \dots, X_n\}$ , such that  $H_m(x)$ , the information or entropy of the subsample becomes as close to  $H_n(x)$  i.e. the entropy of the original sample. As mentioned before, two entropy criteria will be discussed i.e. Shannon and Rényi. The Shannon or differential entropy  $H_S(X)$  is defined by

$$\begin{aligned} H_S(X) &= \mathbf{E}[-\log f(X)] \\ &= - \int_{\mathbb{R}^d} f(x) \log(f(x)) dx, \end{aligned} \quad (4.10)$$

and the Rényi entropy  $H_R^{(q)}(x)$  of order  $q$  is defined as

$$H_R^{(q)}(x) = \frac{1}{1-q} \log \int f(x)^q dx, \quad (4.11)$$

with  $q > 0$ ,  $q \neq 1$ . In order to compute both entropy criteria it can be seen that an estimate of the density  $f$  is required.

The (multivariate) density function  $f$  can be estimated by the (multivariate) kernel density estimator (Silverman, 1986; Scott, 1992)

$$\hat{f}(x_1, \dots, x_d) = \frac{1}{n \prod_{j=1}^d h_j} \sum_{i=1}^n \left\{ \prod_{j=1}^d K \left( \frac{x_i - X_{ij}}{h_j} \right) \right\},$$

where  $h_j$  denotes the bandwidth for each dimension  $j$  and the kernel  $K : \mathbb{R} \rightarrow \mathbb{R}^+$  satisfies  $\int_{\mathbb{R}} K(u) du = 1$ . For  $d > 1$ , the same (univariate) kernel is used in each dimension but with a different smoothing parameter (bandwidth) for each dimension. The point  $X_{ij}$  is the  $ij^{\text{th}}$  entry of the given data matrix  $(X_1, \dots, X_n)^T \in \mathbb{R}^{n \times d}$ . Another possibility to estimate  $f(x)$  is by using the general multivariate kernel estimator (Scott, 1992) given by

$$\hat{f}(x) = \frac{1}{n|D|} \sum_{i=1}^n K \{D^{-1}(x - X_i)\}, \quad (4.12)$$

where  $|\cdot|$  denotes the determinant,  $D$  is a non-singular matrix of the form  $D = \text{diag}(h_1, \dots, h_d)$  and the kernel  $K : \mathbb{R}^d \rightarrow \mathbb{R}^+$  satisfies  $\int_{\mathbb{R}^d} K(u) du = 1$ . In what follows the general multivariate kernel estimator (4.12) will be used. Calculation of the bandwidths  $h_1, \dots, h_d$  will be discussed in the next two paragraphs.

When the Shannon (differential) entropy, given by (4.10), is used along with the kernel density estimate  $\hat{f}(x)$ , the estimation of the entropy  $H_S(x)$  becomes very complicated. However, Rényi's entropy of order  $q = 2$ , denoted by  $H_R^{(2)}(x)$ , (also called quadratic Rényi entropy) leads to a simpler estimate of entropy, see (4.11). The Shannon entropy can be viewed as one member of the Rényi's entropy family, because  $\lim_{q \rightarrow 1} H_R^{(q)}(x) = H_S(x)$ . Although Shannon's entropy is the only one which possesses properties such as continuity, symmetry, extremal property, recursivity and additivity for an information measure, Rényi's entropy family is equivalent with respect to entropy maximization (Rényi, 1961, 2007). In real problems, the choice of information measure depends on other requirements such as ease of implementation. Combining (4.12) and (4.11), Rényi's quadratic entropy estimator, based on  $m$  prototype vectors and setting  $q = 2$ , becomes

$$\hat{H}_R^{(2)}(X) = -\log \frac{1}{m^2|D|^2} \sum_{k=1}^m \sum_{l=1}^m K \left\{ \left( D\sqrt{2} \right)^{-1} (X_k - X_l) \right\}. \quad (4.13)$$

We choose a fixed size  $m$  ( $m \ll n$ ) for a working set of data points (prototype vectors) and actively select points from the pool of training input samples as a candidate for the working set (PVs). In the working set, a point is randomly selected and replaced by a randomly selected point from the training input sample if the new point improves Rényi's quadratic entropy criterion. This leads to the following active selection algorithm as introduced in Suykens et al. (2002). In the classification setting, Algorithm 1 can be used in a stratified sampling scheme.

**Algorithm 1** Active prototype vector selection

- 
- 1: Given a training set  $\mathcal{D}_n = \{(X_1, Y_1), \dots, (X_n, Y_n)\}$ , choose a working set of prototype vectors  $\mathcal{W}_m \subset \mathcal{D}_n$  of size  $m$  at random
  - 2: Randomly select a sample point  $X^* \in \mathcal{W}_m$ , and  $X^+ \in \mathcal{D}_n$ , swap  $(X^*, X^+)$
  - 3: **if**  $\hat{H}_R^{(2)}(X_1, \dots, X_{m-1}; X^+) > \hat{H}_R^{(2)}(X_1, \dots, X_i^*, \dots, X_m)$  **then**
  - 4:    $X^+ \in \mathcal{W}_m$  and  $X^* \notin \mathcal{W}_m, X^* \in \mathcal{D}_n$
  - 5: **else**
  - 6:    $X^+ \notin \mathcal{W}_m$  and  $X^* \in \mathcal{W}_m, X^* \in \mathcal{D}_n$
  - 7: **end if**
  - 8: Calculate  $\hat{H}_R^{(2)}(X)$  for the present  $\mathcal{W}_m$
  - 9: Stop if the change in entropy value (4.13) is small
- 

**Remark** *In our kernel entropy criterion, we have used the Gaussian kernel. However, it can be shown that better kernel entropy estimates can be obtained by using the naive (uniform or step) kernel if one is interested in the numerical value of the entropy. Paninski and Yajima (2008) provide a kernel entropy estimator whose error term may be bounded by a term which goes to zero if the kernel bandwidth scales as  $1/n$ . Therefore, accurate density estimates are not required for accurate kernel entropy estimates. In fact it is a good idea when estimating entropy to sacrifice some accuracy in the quality of the corresponding density estimate i.e. to undersmooth. Paninski and Yajima (2008) also show that a uniformly consistent kernel entropy estimator exists if  $nh \rightarrow 0$  sufficiently slowly. Finally, it was already observed in Beirlant et al. (1997) that consistent density estimates are not required for consistent entropy estimates. Mnatsakanov et al. (2008) and Leonenko et al. (2008a,b) showed the asymptotic unbiasedness and consistency of the Shannon, Rényi and Tsallis entropy, under minimal assumptions on the density, based on  $k$ -nearest neighbors. Leonenko and Seleznev (2010) proved the asymptotic normality of these estimators and illustrated applications areas in mathematical statistics and computer science. Numerical implementations and speed-ups regarding these type of estimators, based on  $k$ -nearest neighbors, are given by Vejmelka and Hlaváčková-Schindler (2007).*

### 4.3.2 Bandwidth Selection for Density Estimation

The most popular non-parametric density estimate  $\hat{f}$  of a density  $f$  based on a random sample  $X_1, \dots, X_n$  is given by (Rosenblatt, 1956; Parzen, 1962)

$$\hat{f}(x) = \frac{1}{nh^d} \sum_{k=1}^n K\left(\frac{x - X_k}{h}\right).$$

Efficient use of kernel density estimation requires the optimal selection of the bandwidth of the kernel. A wide variety of methods to select the bandwidth

$h$  in the case of kernel density estimation are available e.g. least squares cross-validation (Rudemo, 1982; Bowman, 1984; Silverman, 1986; Li and Racine, 2006), biased cross-validation (Scott and Terrell, 1987), bootstrap bandwidth selector (Hall et al., 1999; Li and Racine, 2006), regression-based bandwidth selector (Härdle, 1991; Fan and Gijbels, 1996), double kernel method (Devroye, 1989; Devroye and Lugosi, 2001), plug-in methods (Silverman, 1986; Li and Racine, 2006; Raykar and Duraiswami, 2006), normal reference rule of thumb (Silverman, 1986; Scott, 1992) and the test graph method (Silverman, 1978, 1986).

However, since large data sets are considered, computational aspects of the selection of the bandwidth should not be neglected. Therefore, only the normal reference rule of thumb and plug-in methods can be considered. In what follows only the plug-in method will be discussed. The selection of the smoothing parameter is based on choosing  $h$  to minimize a kernel-based estimate of the mean integrated squared error (MISE)

$$\begin{aligned}
 \text{MISE}(\hat{f}) &= \mathbf{E} \int [\hat{f}(x) - f(x)]^2 dx \\
 &\stackrel{\text{Fubini}}{=} \int \mathbf{E} [\hat{f}(x) - f(x)]^2 dx \\
 &= \int \text{MSE}(\hat{f}(x)) dx \\
 &= \int \text{bias}^2(\hat{f}(x)) dx + \int \mathbf{Var}(\hat{f}(x)) dx. \tag{4.14}
 \end{aligned}$$

Let  $X_1, \dots, X_n \stackrel{\text{i.i.d.}}{\sim} X$  and under necessary conditions on  $f$  and  $K$ , an asymptotic approximation for bias of  $\hat{f}(x)$  is given by

$$\begin{aligned}
 \mathbf{E}[\hat{f}(x)] &= \frac{1}{h} \int K\left(\frac{x-y}{h}\right) f(y) dy \\
 &= \int K(u) f(x-uh) du, \quad \text{where } u = \left(\frac{x-y}{h}\right) \\
 &= f(x) \int K(u) du - f'(x)h \int uK(u) du + \frac{1}{2}f''(x)h^2 \int u^2K(u) du + o(h^2).
 \end{aligned}$$

If

$$K \geq 0, \quad \int K(u) du = 1, \quad \int uK(u) du = 0, \quad 0 < \int u^2K(u) du < \infty,$$

and the underlying unknown density  $f$  has continuous derivatives of all orders required, then an approximations for the bias is given by

$$\mathbf{E}[\hat{f}(x)] - f(x) = \frac{1}{2}f''(x)h^2 \int u^2K(u) du + o(h^2).$$

The variance of  $\hat{f}(x)$  is given by

$$\mathbf{Var}[\hat{f}(x)] = \frac{1}{n} \left[ \frac{1}{h^2} \mathbf{E} \left[ K^2 \left( \frac{x-X}{h} \right) \right] - \left\{ \frac{1}{h} \mathbf{E} \left[ K \left( \frac{x-X}{h} \right) \right] \right\}^2 \right]. \quad (4.15)$$

Since

$$\begin{aligned} \frac{1}{h^2} \mathbf{E} \left[ K^2 \left( \frac{x-X}{h} \right) \right] &= \frac{1}{h} \int K^2(u) f(x-uh) du \\ &= \frac{1}{h} f(x) \int K^2(u) du - f'(x) \int u K^2(u) du + o(1) \end{aligned}$$

under the same assumptions as before, the variance is given by

$$\mathbf{Var}[\hat{f}(x)] = \frac{1}{nh} f(x) \int K^2(u) du + o(n^{-1}h^{-1}). \quad (4.16)$$

Hence, the asymptotic approximation for  $\text{MISE}(\hat{f})$  is given by plugging (4.15) and (4.16) in (4.14), yielding

$$\text{MISE}(\hat{f}) = \frac{1}{4} h^4 \mu_2^2(K) \int (f''(x))^2 dx + \frac{1}{nh} R(K) + o(h^4 + \frac{1}{nh}),$$

with  $\mu_2(K) = \int u^2 K(u) du$  and  $R(K) = \int K^2(u) du$ . The asymptotic MISE bandwidth  $h_{\text{AMISE}}$  is given by minimizing the asymptotic MISE

$$h_{\text{AMISE}} = \arg \min_h \text{MISE}(\hat{f}). \quad (4.17)$$

The result of (4.17) is given by

$$h_{\text{AMISE}} = \left[ \frac{R(K)}{\mu_2^2(K) R(f'')} \right]^{1/5} n^{-1/5}. \quad (4.18)$$

However, the above expression cannot be used directly since  $R(f'')$  depends on the second derivative of the density  $f$ . An estimator of the functional  $R(f^{(r)})$  using a kernel density derivative estimate for  $f^{(r)}$ , with bandwidth  $g$ , is given by

$$\hat{R}(f^{(r)}) = \frac{1}{n^2 g_{\text{AMISE}}^{r+1}} \sum_{i=1}^n \sum_{j=1}^n K^{(r)} \left( \frac{X_i - X_j}{g_{\text{AMISE}}} \right). \quad (4.19)$$

The optimal bandwidth  $g_{\text{AMISE}}$  for estimating the density functional is given in Theorem 4.1. Note that it is not necessary to take the same kernel functions for density estimation and for the estimation of the functional  $R(f^{(r)})$ , see e.g. Scott et al. (1977).

**Theorem 4.1 (Wand and Jones, 1995)** *The asymptotic mean integrated squared error (AMISE) optimal bandwidth  $g_{\text{AMISE}}$  for (4.19) is given by*

$$\hat{g}_{\text{AMISE}} = \left( \frac{-2K^{(r)}(0)}{\mu_2(K) \hat{R}(f^{(r+2)})} \right)^{1/(r+3)} n^{-1/(r+3)}.$$

### 4.3.3 Solve-the-Equation Plug-In Method

One of the most successful methods for bandwidth selection for kernel density estimation is the *solve-the-equation plug-in method* (Sheather and Jones, 1991; Jones et al., 1996). The basic idea is to write the AMISE optimal bandwidth (4.18) as follows

$$\hat{h}_{\text{AMISE}} = \left( \frac{R(K)}{\mu_2^2(K) \hat{R}(f^{(4)}, \rho(\hat{h}_{\text{AMISE}}))} \right)^{1/5} n^{-1/5}, \quad (4.20)$$

where  $\hat{R}(f^{(4)}, \rho(\hat{h}_{\text{AMISE}}))$  is an estimate of  $R(f^{(4)})$  using the pilot bandwidth  $\rho(h_{\text{AMISE}})$ . Note that this bandwidth to estimate the density functional (4.19) is different from the bandwidth  $h_{\text{AMISE}}$  used for kernel density estimation. Based on Theorem 4.1 the bandwidth  $\hat{g}_{\text{AMISE}}$  for  $R(f^{(4)})$  is given by

$$\hat{g}_{\text{AMISE}} = \left( \frac{-2K^{(4)}(0)}{\mu_2(K) \hat{R}(f^{(6)}, \rho(\hat{g}_{\text{AMISE}}))} \right)^{1/7} n^{-1/7}.$$

Using (4.18) and substituting for  $n$ ,  $g_{\text{AMISE}}$  can be written as a function of the bandwidth  $h_{\text{AMISE}}$  for kernel density estimation

$$\hat{g}_{\text{AMISE}} = \left( \frac{-2K^{(4)}(0) \mu_2(K) \hat{R}(f^{(4)}, \hat{h}_1)}{R(K) \hat{R}(f^{(6)}, \hat{h}_2)} \right)^{1/7} \hat{h}_{\text{AMISE}}^{5/7},$$

where  $\hat{R}(f^{(4)}, \hat{h}_1)$  and  $\hat{R}(f^{(6)}, \hat{h}_2)$  are estimates of  $R(f^{(4)}, h_1)$  and  $R(f^{(6)}, h_2)$  using bandwidths  $h_1$  and  $h_2$  respectively. The bandwidths are chosen such that they minimize the AMISE and are given by Theorem 4.1

$$\hat{h}_1 = \left( \frac{-2K^{(4)}(0)}{\mu_2(K) \hat{R}(f^{(6)})} \right)^{1/7} n^{-1/7} \quad \text{and} \quad \hat{h}_2 = \left( \frac{-2K^{(6)}(0)}{\mu_2(K) \hat{R}(f^{(8)})} \right)^{1/9} n^{-1/9}, \quad (4.21)$$

where  $\hat{R}(f^{(6)})$  and  $\hat{R}(f^{(8)})$  are estimates of  $R(f^{(6)})$  and  $R(f^{(8)})$ .

This of course also reveals the problem of how to choose  $r$ , the number of stages. As  $r$  increases the variability of this bandwidth selector will increase, but it becomes less biased since the dependence on the normal reference rule diminishes. Theoretical considerations by Hall and Marron (1987) and Park and Marron (1992) favour taking  $r$  to be at least 2, with  $r = 2$  being a common choice.

If  $f$  is a normal density with variance  $\sigma^2$  then, according to Wand and Jones (1995),  $R(f^{(6)})$  and  $R(f^{(8)})$  can be calculated exactly. An estimator of  $R(f^{(r)})$

will use an estimate  $\hat{\sigma}^2$  of the variance. An estimator for  $R(f^{(6)})$  and  $R(f^{(8)})$ , in case of a normal density with variance  $\sigma^2$ , is given by

$$\hat{R}(f^{(6)}) = \frac{-15}{16\sqrt{\pi}} \hat{\sigma}^{-7} \quad \text{and} \quad \hat{R}(f^{(8)}) = \frac{105}{32\sqrt{\pi}} \hat{\sigma}^{-9}. \quad (4.22)$$

The main computational bottleneck is the estimation of the kernel density derivatives  $R(f^{(r)})$  which is of  $O(n^2)$ . A method for fast evaluation of these kernel density derivatives  $R(f^{(r)})$  is proposed in Raykar and Duraiswami (2006). This method is based on the Taylor expansion of the Gaussian and hence adopts the main idea of the Improved Fast Gauss Transform (IFGT) (Yang et al., 2003). IFGT reduces the complexity to  $O(n)$ . However, the constant factor in  $O(n)$  grows exponentially with increasing dimensionality  $d$ , which makes the algorithm impractical in higher dimensions i.e.  $d > 9$ .

The two stage solve-the-equation plug-in method using a Gaussian kernel is given in Algorithm 2. A general overview of IFGT with applications to machine learning can be found in Raykar and Duraiswami (2007).

---

**Algorithm 2** solve-the-equation plug-in method

---

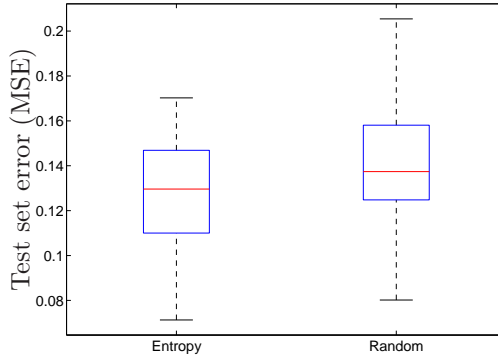
- 1: Compute an estimate  $\hat{\sigma}$  of the standard deviation.
  - 2: Estimate density functionals  $\hat{R}(f^{(6)})$  and  $\hat{R}(f^{(8)})$  using (4.22).
  - 3: Estimate density functionals  $\hat{R}(f^{(4)})$  and  $\hat{R}(f^{(6)})$  with bandwidths  $\hat{h}_1$  and  $\hat{h}_2$  using (4.21) and (4.19).
  - 4: The optimal bandwidth is the solution to the nonlinear equation (4.20). This equation can be solved by using e.g. Newton-Raphson method.
- 

The previously discussed method is able, for a given univariate data set, to compute the bandwidth. However, in the multivariate case one needs to compute several bandwidths. One can then proceed as follows. To obtain the bandwidth matrix  $D = \text{diag}(h_1, \dots, h_d)$ , one computes a bandwidth  $h_i$ ,  $i = 1, \dots, d$ , for each dimension according to Algorithm 2. This bandwidth matrix  $D$  can then be used in (4.13) to compute the entropy estimate of multivariate sample.

#### 4.3.4 Maximizing Rényi Entropy vs. Random Sampling

It is possible to compare the performance on test between models estimated with a random prototype vector selection versus the same models estimated with quadratic Rényi entropy based prototype vector selection. In order to compare both performances on test we use the UCI Boston Housing data set (this data set is publicly available at <http://kdd.ics.uci.edu/>). 168 test data points were used and the number of prototype vectors was set to  $m = 200$ . The test is randomly

selected in each run. Each model is tuned via 10-fold CV for both selection criteria. Figure 4.1 shows the comparison for the results based on 100 runs. Table 4.1 shows the average MSE and the standard deviation of the MSE. These results show that using the entropy based criterion yields a lower mean and dispersion value on test. Similar results were also obtained for different data sets.



**Figure 4.1:** Boxplot of the MSE on test (100 runs) for models estimated with entropy based and random selection of prototype vectors.

**Table 4.1:** Comparison of the mean and standard deviation of the MSE on test for the Boston Housing data set using  $m = 200$  over 100 randomizations.

Selection method	Average MSE	standard deviation MSE
Entropy	0.1293	0.0246
Random	0.1433	0.0323

Maximizing entropy criteria has been a widely studied area, see e.g. Cover and Thomas (1991). The density which maximizes the entropy on a closed interval without imposing additional moment constraints is the uniform density on that closed interval. Theorem 4.2 states this result for the one dimensional case.

**Theorem 4.2** *The Rényi entropy on a closed interval  $[a, b]$  with  $a, b \in \mathbb{R}$  and no additional moment constraints is maximized for the uniform density  $1/(b - a)$ .*

PROOF. For  $q > 0$  and  $q \neq 1$ , maximizing the Rényi entropy with density  $f$

$$\frac{1}{1 - q} \log \int_a^b f^q(x) dx$$



is equivalent to minimizing

$$\int_a^b f^q(x) dx,$$

since  $\frac{1}{1-q}$  is always negative and logarithm is a monotone function. By the reverse of Jensen's inequality and noting that  $f$  is a density

$$\frac{1}{b-a} \int_a^b f^q(x) dx \geq \left( \frac{1}{b-a} \int_a^b f(x) dx \right)^q = \frac{1}{(b-a)^q}.$$

Therefore, it follows that

$$\int_a^b f^q(x) dx \geq (b-a)^{1-q}$$

for  $\int_a^b f(x) dx = 1$ . This lower bound is reached if

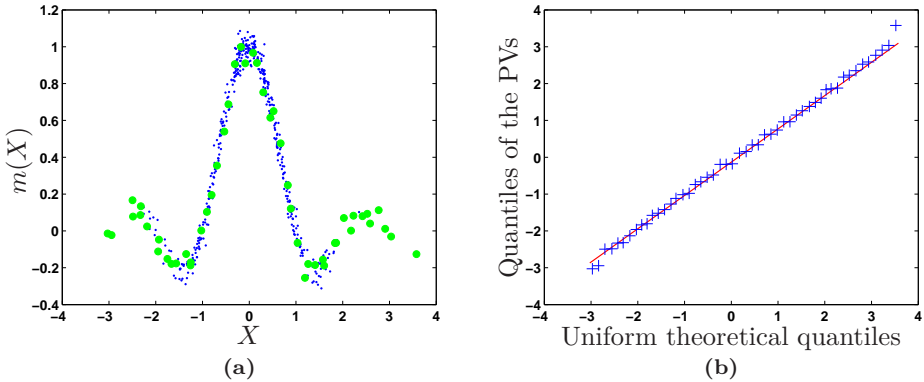
$$f(x) = \frac{1}{b-a} \text{ on } [a,b].$$

□

Therefore, using a maximum entropy strategy to select the position of PVs will result in a uniform subset over the input space (the selection only considers the inputs and not the outputs), at least in theory. This is illustrated by means of the following toy example. Let  $m(X) = \text{sinc}(X) + \varepsilon$  where  $X \sim \mathcal{N}(0,1)$  and  $\varepsilon \sim \mathcal{N}(0,0.1^2)$ . The sample size was taken to be 500. Maximizing the quadratic Rényi entropy (4.13) was chosen as criterion to select 50 PVs. The kernel entropy bandwidth  $h = 0.2634$  was determined by the solve-the-equation plug-in method (4.20). The results are shown in Figure 4.2. From the Q-Q plot one can observe that the selected subset has a uniform distribution over the input space.

## 4.4 Selecting the number of prototype vectors

An important task in this framework is to determine the number of PVs  $m \in \mathbb{N}_0$  used in the FS-LSSVM model. Existing methods (Smola and Schölkopf, 2000; Keerthi et al., 2006; Jiao et al., 2007; Zhao and Sun, 2009) select the number of PVs based on a greedy approach. One disadvantage of these type of methods is that they are time consuming, hence making them infeasible for large scale data sets. In Smola and Schölkopf (2000) and Keerthi et al. (2006) an additional parameter is introduced in the subset selection process which requires tuning. This parameter determines the number of random PVs to try outside the already selected subset. Keerthi et al. (2006) pointed out that there is no universal answer to set this parameter because the answer would depend on the cost associated with the



**Figure 4.2:** (a) Illustration of the function  $m(X)$  and the selected prototype vectors (big dots); (b) Q-Q plot of the selected prototype vectors ( $x$ -coordinate) versus a uniform distribution. The data falls clearly on the the straight line, confirming that the selected subset has a uniform distribution over the input space.

computation of the kernel function, on the number of selected PVs and on the number of training points.

A second class of methods to select the PVs is by constructing a basis in feature space (Baudat and Anouar, 2001). This method was also used by Cawley and Talbot (2002). The main idea of the method is to minimize the normalized Euclidean distance between the position of a data item in feature space and its optimal reconstruction using the set of basis vectors (PVs). This method is computationally heavy since it involves numerous matrix-matrix products and matrix inverses.

A third class of method is based on matrix algebra to obtain the number of PVs. Valyon and Horváth (2004) obtain PVs by bringing the kernel matrix to the reduced row echelon form (Golub and Van Loan, 1996). The total complexity of the method is given by  $\frac{1}{3}m^3 + m^2(n+1) + n^2$ . This method can become intractable for large data sets. Abe (2007) uses an incomplete Cholesky factorization to select the PVs. The number of PVs are controlled by an extra tuning parameter that can be determined by CV. A benefit of this method is that it does not require storage of the complete kernel matrix into the memory, which can be problematic when data sets are large, but the factorization can be done incrementally.

A fourth class is based on error bounds. Zhang et al. (2008) have used a  $k$ -means clustering algorithm to select the PVs (i.e.  $m$ ) for the Nyström approximation (4.6). However, the bound in approximating the full kernel matrix using a low rank approximation (Nyström method) appears not to be tight. Therefore, the number

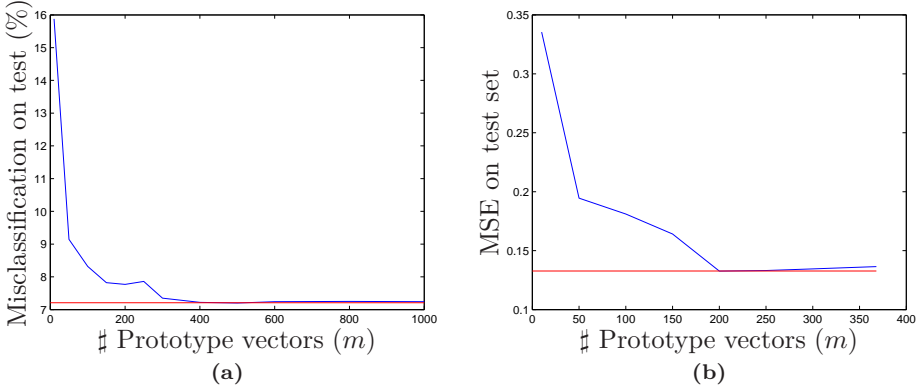
of PVs can be expected to be relatively large compared to  $n$ , hence making this strategy not well suited for large data sets.

More recently, an optimization scheme for computing sparse approximate solutions of over-determined linear systems was proposed by Karsmakers et al. (2010) called Sparse Conjugate Directions Pursuit (SCDP). SCDP aims to construct a solution using only a small number of nonzero (i.e. non-sparse) coefficients. The main idea is to build up iteratively a conjugate set of vectors (PVs) of increasing cardinality, by solving in each iteration a small linear subsystem. By exploiting the structure of this conjugate basis, an algorithm is found (i) converging in at most  $d$ -iterations for  $d$ -dimensional systems, (ii) with computational complexity close to the classical conjugate gradient algorithm, and (iii) which is especially efficient when a few iterations suffice to produce a good approximation.

Our proposed method selects the PVs by maximizing the quadratic Rényi entropy (4.13). This algorithm requires the entropy kernel bandwidth(s). The computational complexity associated with determining  $d$  bandwidths is roughly  $nd$  (Raykar and Duraiswami, 2006). One only has to calculate the kernel matrix associated with the PVs which has a computational cost of  $m^2$ . The entropy value is then almost given by the sum of all elements of this kernel matrix. Each time when the set of PVs is altered by using the active selection strategy the entropy value of the new set can be simply obtained by updating the previous value. Hence, in this strategy the kernel matrix associated with the PVs has to be calculated only once and can then be removed from the memory.

In theory we could gradually increase the number of PVs till  $m = n$ . Naturally it would be too computationally expensive, but it gives an insight of how an increasing amount of prototype vectors will influence the performance (on test) of a classifier or regression estimate. Consider the Spam data set (1533 data points are randomly selected as test set) and the Boston Housing data set (168 data points are randomly selected as test set). In these examples the number of PVs  $m$  are gradually increased and the performance on test data is determined for each of the chosen prototype vector sizes. Figure 4.3 shows the number of prototype vectors of the FS-LSSVM as a function of the performance on test data for both data sets. The straight line is the LS-SVM estimate on the same data sets and serves as a baseline comparison. These results indicate that only a percentage of the total number of data points is required to obtain a performance on test which is equal to the LS-SVM estimate.

In practice, however, due to time constraints and computational burden, it is impossible to gradually increase the number of prototype vectors till e.g.  $m = n$ . Therefore, we propose a simple heuristic in order to obtain a rough estimate of the number of prototype vectors to be used in the FS-LSSVM model. Choose  $k$  different values for the number of prototype vectors, say  $k = 5$ . Determine  $k$  FS-LSSVM models and calculate the performance on test of each model. Choose



**Figure 4.3:** Number of prototype vectors in function of the performance on a test set. The straight line is the LS-SVM estimate and serves as a baseline comparison. (a) Spam data (binary classification); (b) Boston Housing data (regression).

as final  $m$ , the number of prototype vector of the model which has the best performance on test data. Also note that in this way not always the sparsest model is selected, but it reduces computation time.

In the FS-LSSVM framework the number of PVs,  $m$ , is a tuning parameter but the choice is not very crucial as can be seen in Figure 4.3. This is however an inherent drawback of the method in contrast to SVM where the number of SVs follow from solving a convex QP problem.

## 4.5 Fast $v$ -fold Cross-Validation for FS-LSSVM

When considering large data sets, it is also important to have a fast and accurate algorithm for cross-validation (see also Chapter 3). In what follows we propose a fast algorithm for  $v$ -fold CV based on a simple updating scheme for FS-LSSVM. Depending whether the extended feature matrix  $\hat{\Phi}_e$  (4.9) can be stored completely into the memory or not, two version of the algorithm will be discussed.

### 4.5.1 Extended Feature Matrix Can Fit Into Memory

The algorithm is based on the fact that the extended feature matrix  $\hat{\Phi}_e$  (4.9) has to be calculated only once instead of  $v$  times. Unlike An et al. (2007) and Ying and Keong (2004), the algorithm is not based on one full matrix inverse because of the complexity  $O(m+1)^3$  of this operation.

Given a data set  $\mathcal{D}_n$ . At each fold of the cross-validation, the  $v^{\text{th}}$  group is left out for validation and the remaining is for training. Recall that the extended feature matrix  $\hat{\Phi}_e \in \mathbb{R}^{n \times (m+1)}$  is given by

$$\hat{\Phi}_e = \left( \begin{array}{ccc|c} \hat{\varphi}_1(X_{tr,1}) & \cdots & \hat{\varphi}_m(X_{tr,1}) & 1 \\ \vdots & \ddots & \vdots & \vdots \\ \hat{\varphi}_1(X_{tr,n_{tr}}) & \cdots & \hat{\varphi}_m(X_{tr,n_{tr}}) & 1 \\ \hline \hat{\varphi}_1(X_{val,1}) & \cdots & \hat{\varphi}_m(X_{val,1}) & 1 \\ \vdots & \ddots & \vdots & \vdots \\ \hat{\varphi}_1(X_{val,n_{val}}) & \cdots & \hat{\varphi}_m(X_{val,n_{val}}) & 1 \end{array} \right) = \left( \begin{array}{c|c} \hat{\Phi}_{tr} & \mathbf{1}_{tr} \\ \hline \hat{\Phi}_{val} & \mathbf{1}_{val} \end{array} \right), \quad (4.23)$$

where  $\mathbf{1}_{tr} = (1, \dots, 1)^T$ ,  $\mathbf{1}_{val} = (1, \dots, 1)^T$ ,  $X_{tr,j}$  and  $X_{val,j}$  denote the  $j^{\text{th}}$  element of the training data and validation data respectively.  $n_{tr}$  and  $n_{val}$  are the number of data points in the training data set and validation set respectively such that  $n_{tr} + n_{val} = n$ . Also set  $A = \hat{\Phi}_e^T \hat{\Phi}_e + \frac{I_{m+1}}{\gamma}$  and  $c = \hat{\Phi}_e^T Y$ . At each fold of the cross-validation, the  $v$ -th group is left out for validation and the remaining is for training. So in the  $v$ -th iteration

$$A_v \begin{pmatrix} \tilde{w} \\ b \end{pmatrix} = c_v, \quad (4.24)$$

where  $A_v$  is a square matrix with the same dimension as  $A$  but modified from  $A$  by taking only the training data to build it. The motivation is to get  $A_v$  from  $A$  with a few simple steps instead of computing  $A_v$  in each fold from scratch. Using (4.8) and (4.23), the following holds

$$\begin{aligned} \hat{\Phi}_e^T \hat{\Phi}_e + \frac{I_{m+1}}{\gamma} &= \left( \begin{array}{c|c} \hat{\Phi}_{tr}^T & \hat{\Phi}_{val}^T \\ \hline \mathbf{1}_{tr}^T & \mathbf{1}_{val}^T \end{array} \right) \left( \begin{array}{c|c} \hat{\Phi}_{tr} & \mathbf{1}_{tr} \\ \hline \hat{\Phi}_{val} & \mathbf{1}_{val} \end{array} \right) + \frac{I_{m+1}}{\gamma} \\ &= \left( \begin{array}{c|c} \hat{\Phi}_{tr}^T \hat{\Phi}_{tr} + \frac{I_m}{\gamma} & \hat{\Phi}_{tr}^T \mathbf{1}_{tr} \\ \hline \mathbf{1}_{tr}^T \hat{\Phi}_{tr} & \mathbf{1}_{tr}^T \mathbf{1}_{tr} + \frac{1}{\gamma} \end{array} \right) + \left( \begin{array}{c|c} \hat{\Phi}_{val}^T \hat{\Phi}_{val} & \hat{\Phi}_{val}^T \mathbf{1}_{val} \\ \hline \mathbf{1}_{val}^T \hat{\Phi}_{val} & \mathbf{1}_{val}^T \mathbf{1}_{val} \end{array} \right) \\ &= \left( \hat{\Phi}_{e,tr}^T \hat{\Phi}_{e,tr} + \frac{I_{m+1}}{\gamma} \right) + \left( \begin{array}{c|c} \hat{\Phi}_{val}^T & \\ \hline \mathbf{1}_{val}^T & \end{array} \right) \left( \begin{array}{c|c} \hat{\Phi}_{val} & \\ \hline \mathbf{1}_{val} & \end{array} \right), \end{aligned}$$

where  $\hat{\Phi}_{e,tr} = \left( \begin{array}{c|c} \hat{\Phi}_{tr} & \\ \hline \mathbf{1}_{tr} & \end{array} \right)$  is the extended feature matrix of the training data and hence

$$\hat{\Phi}_{e,tr}^T \hat{\Phi}_{e,tr} + \frac{I_{m+1}}{\gamma} = \left( \hat{\Phi}_e^T \hat{\Phi}_e + \frac{I_{m+1}}{\gamma} \right) - \left( \begin{array}{c|c} \hat{\Phi}_{val}^T & \\ \hline \mathbf{1}_{val}^T & \end{array} \right) \left( \begin{array}{c|c} \hat{\Phi}_{val} & \\ \hline \mathbf{1}_{val} & \end{array} \right).$$

This results in

$$A_v = A - \left( \begin{array}{c} \hat{\Phi}_{val}^T \\ 1_{val}^T \end{array} \right) \left( \hat{\Phi}_{val} \mid 1_{val} \right). \quad (4.25)$$

The most time consuming step in (4.25) is the calculation of matrix  $A$ . However, this calculation needs to be performed only once. The second term in (4.25) does not require complete recalculation since  $\hat{\Phi}_{val}$  can be extracted from  $\hat{\Phi}_e$ , see (4.23). A similar result holds for  $c_v$

$$\begin{aligned} c &= \hat{\Phi}_e^T Y \\ &= \left( \begin{array}{c|c} \hat{\Phi}_{tr}^T & \hat{\Phi}_{val}^T \\ \hline 1_{tr}^T & 1_{val}^T \end{array} \right) \left( \begin{array}{c} Y_{tr} \\ Y_{val} \end{array} \right) \\ &= \left( \frac{\hat{\Phi}_{tr}^T}{1_{tr}^T} \right) Y_{tr} + \left( \frac{\hat{\Phi}_{val}^T}{1_{val}^T} \right) Y_{val} \\ &= c_v + \left( \frac{\hat{\Phi}_{val}^T}{1_{val}^T} \right) Y_{val}, \end{aligned}$$

thus

$$c_v = c - \left( \frac{\hat{\Phi}_{val}^T}{1_{val}^T} \right) Y_{val}. \quad (4.26)$$

In each fold one has to solve the linear system (4.24). This leads to Algorithm 3 for fast  $v$ -fold cross-validation.

---

**Algorithm 3** Fast  $v$ -fold cross-validation for FS-LSSVM

---

- 1: Calculate the matrix  $\hat{\Phi}_e$  (for all data using the Nyström approximation (4.6)) and  $A = \hat{\Phi}_e^T \hat{\Phi}_e + \frac{I_{m+1}}{\gamma}$ .
  - 2: Split data randomly into  $v$  disjoint sets of nearly equal size.
  - 3: Compute in each fold  $A_v$  and  $c_v$  using (4.25) and (4.26) respectively.
  - 4: Solve the linear system (4.24).
  - 5: Compute the residuals in each fold  $\hat{e}_{v,i} = Y_{val,i} - (\hat{w}^T \hat{\varphi}(X_{val,i}) + \hat{b})$  in each fold.
  - 6: Choose an appropriate loss function to assess performance (e.g. MSE).
- 

The literature describes other fast implementations for CV, see e.g. the methods of Cawley and Talbot (2004) and Ying and Keong (2004) for leave-one-out CV (LOO-CV) and An et al. (2007) for  $v$ -fold CV. The method of choice greatly depends on the number of folds used for CV. Table 4.2 gives an overview of which method can be best used with different type of folds.

**Table 4.2:** Summary of different implementations of CV.

implementation	# folds
Cawley and Talbot (2004)	$n$
An et al. (2007)	$> 20$
proposed	3-20

#### 4.5.2 Extended Feature Matrix Cannot Fit Into Memory

The fast  $v$ -fold CV algorithm for FS-LSSVM (Algorithm 3) is based on the fact that the extended feature matrix (4.9) can fit into the memory. In order to overcome this problem we propose to calculate the extended feature matrix  $\hat{\Phi}_e$  in a number of  $S$  blocks. In this way, the extended feature matrix  $\hat{\Phi}_e$  does not need to be stored completely into the memory. Let  $l_s$ , with  $s = 1, \dots, S$  denote the length of the  $s$ -th block and also  $\sum_{s=1}^S l_s = n$ . The matrix  $\hat{\Phi}_e$  can be written as follows

$$\hat{\Phi}_e = \begin{pmatrix} \hat{\Phi}_{e,[1]} \\ \vdots \\ \hat{\Phi}_{e,[S]} \end{pmatrix},$$

with  $\hat{\Phi}_{e,[s]} \in \mathbb{R}^{l_s \times (m+1)}$  and the vector  $Y$

$$Y = \begin{pmatrix} Y_{[1]} \\ \vdots \\ Y_{[S]} \end{pmatrix},$$

with  $Y_{[s]} \in \mathbb{R}^{l_s}$ . The matrix  $\hat{\Phi}_{e,[s]}^T \hat{\Phi}_{e,[s]}$  and vector  $\hat{\Phi}_{e,[s]}^T Y_{[s]}$  can be calculated in an updating scheme and stored into the memory since their sizes are  $(m+1) \times (m+1)$  and  $(m+1) \times 1$  respectively.

Also because of the high computational burden, we can validate using a holdout estimate (Devroye et al., 1996). The data sequence  $\mathcal{D}_n$  is split into a training sequence  $\mathcal{D}_{tr} = \{(X_1, Y_1), \dots, (X_t, Y_t)\}$  and a fixed validation sequence  $\mathcal{D}_{val} = \{(X_{t+1}, Y_{t+1}), \dots, (X_{t+l}, Y_{t+l})\}$ , where  $t+l = n$ . Algorithm 4 summarizes the above idea. The idea of calculating  $\hat{\Phi}_e$  in blocks can also be extended to  $v$ -fold CV.

---

**Algorithm 4** Holdout estimate for very large-scale FS-LSSVM
 

---

- 1: Choose a fixed validation sequence  $\mathcal{D}_{val}$ .
  - 2: Divide the remaining data set  $\mathcal{D}_{tr}$  into approximately  $S$  equal blocks such that  $\hat{\Phi}_{e,[s]}$  with  $s = 1, \dots, S$ , calculated by (4.6), can fit into the memory.
  - 3: Initialize matrix  $A_v \in \mathbb{R}^{(m+1) \times (m+1)}$  and vector  $c_v \in \mathbb{R}^{m+1}$ .
  - 4: **for**  $s = 1$  to  $S$  **do**
  - 5:   Calculate matrix  $\hat{\Phi}_{e,[s]}$  for the  $s$ -th block using the Nyström approximation (4.6)
  - 6:    $A_v \leftarrow A_v + \hat{\Phi}_{e,[s]}^T \hat{\Phi}_{e,[s]}$
  - 7:    $c_v \leftarrow c_v + \hat{\Phi}_{e,[s]}^T Y_{[s]}$
  - 8: **end for**
  - 9: Set  $A_v \leftarrow A_v + \frac{I_{m+1}}{\gamma}$ .
  - 10: Solve the linear system (4.24).
  - 11: Compute the residuals  $\hat{e}$  on the fixed validation sequence  $\mathcal{D}_{val}$ .
  - 12: Compute the holdout estimate.
- 

## 4.6 Computational Complexity Analysis and Numerical Experiments on Fast $v$ -fold CV for FS-LSSVM

In this Section, we discuss the complexity of the proposed fast  $v$ -fold CV and present some experimental results compared to a simple implementation of  $v$ -fold CV on a collection of data sets from UCI benchmark repository.

### 4.6.1 Computational Complexity Analysis

The simple implementation of  $v$ -fold CV computes the extended feature matrix (4.9) for each split of data and uses no updating scheme. This is computationally expensive when  $v$  is large (e.g. leave-one-out CV). Note that the complexity of solving a linear system with dimension  $m+1$  is  $\frac{1}{3}(m+1)^3$  (Press et al., 1993) and the complexity of calculating the Nyström approximation (with eigenvalue decomposition of the kernel matrix of size  $m$ ) is  $m^3 + m^2n$ . The total complexity of the proposed method is then given by the sum of the complexities of

- $v$  times solving a linear system of dimensions  $m+1$
- calculating the Nyström approximation + eigenvalue decomposition of the kernel matrix of size  $m$  once
- Matrix matrix product  $\hat{\Phi}_e^T \hat{\Phi}_e$  once.



Hence, the resulting complexity of the proposed method, neglecting lower order terms, is given by  $(\frac{v}{3} + 1)m^3 + 2nm^2$ . In a similar way, the resulting complexity of the simple method is given by  $\frac{4}{3}vm^3 + (2v - 2)nm^2$ . The computational complexity of the proposed method is smaller than the simple method for  $v \geq 2$ . Keeping  $m$  fixed, it is clear that the number of folds has a small influence on the proposed method resulting in a small time increase with increasing number of folds  $v$ . This is in contrast to the simple method where the computational complexity is increasing heavily with increasing folds. On the other hand, consider the number of folds  $v$  fixed and  $m$  variable, a larger time increase is expected with the simple method rather than with the proposed method i.e. the determining factor is  $m^3$ .

## 4.6.2 Numerical Experiments

All the experiments that follow are performed on a PC machine with Intel Core 2 Quad (Q6600) CPU and 3.2 GB RAM under Matlab R2008a for Windows. During the simulations the RBF kernel is used unless mentioned otherwise. To compare efficiency of the proposed algorithm, the experimental procedure, adopted from Mika et al. (1999) and Rätsch et al. (2001), is used where 100 different random training and test splits are defined.

Table 4.3 verifies the computational complexity of the algorithm on the *Concrete Compressive Strength* (publicly available at <http://kdd.ics.uci.edu/> and has 1030 number of instances and 8 attributes.) data set (regression) for various number of prototype vectors (the number of prototype vectors are chosen arbitrarily). From the experiments, it can be seen that the computation time is not very sensitive to the number of folds while this influence is larger in the simple implementation. Both algorithms experience an increasing complexity at an increasing number of prototype vectors. This increase is stronger with the simple implementation. The latter has also a larger standard deviation. The results of Table 4.3 are visualized in Figure 4.4 showing the number of folds as a function of computational time for various number of prototype vectors in the regression case.

Table 4.4 verifies the complexity of the algorithm on the *Magic Gamma Telescope* (this data set is publicly available at <http://kdd.ics.uci.edu/> and has 19020 number of instances and 10 attributes.) data set (binary classification) for various number of prototype vectors (also chosen arbitrarily) and folds. The conclusions are the same as in the regression case. The results of Table 4.4 are visualized in Figure 4.5 showing the number of folds as a function of computational time for various number of prototype vectors in the classification case.

**Table 4.3:** (Regression) The average run time (seconds) over 100 runs of the proposed algorithm compared with the simple implementation for various folds and prototype vectors on the *Concrete Compressive Strength* data set for one pair of fixed tuning parameters. The standard deviation is given within parentheses.

(a) 50 prototype vectors

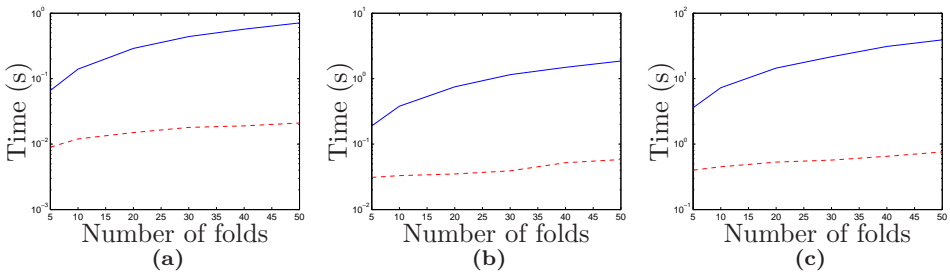
number of folds	5	10	20	30	40	50
simple [s]	0.066 (0.0080)	0.14 (0.0038)	0.29 (0.0072)	0.44 (0.0094)	0.57 (0.0085)	0.71 (0.0079)
optimized [s]	0.009 (0.0013)	0.012 (0.0013)	0.015 (0.0003)	0.018 (0.0004)	0.019 (0.0004)	0.021 (0.0004)

(b) 100 prototype vectors

number of folds	5	10	20	30	40	50
simple [s]	0.19 (0.0060)	0.38 (0.010)	0.75 (0.0140)	1.15 (0.0160)	1.49 (0.0160)	1.86 (0.0160)
optimized [s]	0.031 (0.0060)	0.033 (0.0006)	0.035 (0.0007)	0.039 (0.0001)	0.052 (0.0006)	0.058 (0.0007)

(c) 400 prototype vectors

number of folds	5	10	20	30	40	50
simple [s]	3.60 (0.0400)	7.27 (0.1100)	14.51 (0.1000)	21.62 (0.1100)	31.07 (0.1200)	39.12 (0.1200)
optimized [s]	0.40 (0.0030)	0.45 (0.0020)	0.53 (0.0020)	0.57 (0.0020)	0.65 (0.0050)	0.76 (0.0050)



**Figure 4.4:** Number of folds as a function of computation time (in seconds) for various number of prototype vectors (50, 100, 400) on the Concrete Compressive Strength data set. The full line represents the simple implementation and the dashed line the proposed method.

**Table 4.4:** (Binary Classification) The average run time (seconds) over 100 runs of the proposed algorithm compared with the simple implementation for various folds and prototype vectors on the *Magic Gamma Telescope* data set for one pair of fixed tuning parameters. The standard deviation is given within parentheses.

**(a) 50 prototype vectors**

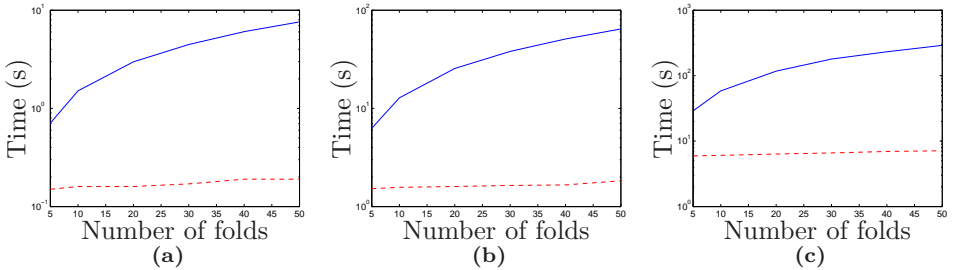
number of folds	5	10	20	30	40	50
simple [s]	0.71 (0.040)	1.51 (0.020)	2.98 (0.026)	4.47 (0.030)	6.05 (0.040)	7.60 (0.039)
optimized [s]	0.15 (0.010)	0.16 (0.006)	0.16 (0.004)	0.17 (0.005)	0.19 (0.005)	0.19 (0.005)

**(b) 300 prototype vectors**

number of folds	5	10	20	30	40	50
simple [s]	6.27 (0.080)	12.80 (0.240)	25.49 (0.270)	38.08 (0.430)	50.94 (0.260)	64.43 (0.420)
optimized [s]	1.52 (0.050)	1.57 (0.020)	1.60 (0.050)	1.64 (0.050)	1.66 (0.040)	1.83 (0.030)

**(c) 700 prototype vectors**

number of folds	5	10	20	30	40	50
simple [s]	29.01 (0.200)	58.56 (0.120)	117.12 (0.160)	179.44 (0.170)	231.59 (0.180)	290.36 (0.710)
optimized [s]	5.95 (0.046)	6.06 (0.090)	6.34 (0.025)	6.60 (0.024)	6.94 (0.032)	7.11 (0.026)



**Figure 4.5:** Number of folds as a function of computation time (in seconds) for various number of prototype vectors (50, 300, 700) on the *Magic Gamma Telescope* data set. The full line represents the simple implementation and the dashed line the proposed method.

## 4.7 Classification and Regression Results

In this Section, we report the application of FS-LSSVM on benchmark data sets (Blake and Merz, 1998) of which a brief description is included in the following paragraph. The performance of the FS-LSSVM is compared to standard SVM and  $\nu$ -SVM (LIBSVM software, Chang and Lin (2001)). Although we focus in this Chapter on large-scale data sets, also smaller data sets will be included for completeness. The randomized test set results are discussed for classification and regression.

### 4.7.1 Description of the Data Sets

All the data sets have been obtained from the publicly accessible UCI benchmark repository (Blake and Merz, 1998). As a preprocessing step, all records containing unknown values are removed from consideration. All given inputs are normalized to zero mean and unit variance.

#### Classification

The following binary data sets were downloaded from <http://kdd.ics.uci.edu/>: Magic Gamma Telescope (**mgt**), Pima Indians Diabetes (**pid**), Adult (**adu**), Spam Database (**spa**) and Forest Covertypes (**ftc**) data set. The main characteristics of these data sets are summarized in Table 4.5. A modification was made from this last data set (Collobert et al., 2002; Hall and Bowyer, 2004): the 7-class classification problem was transformed into a binary classification problem where the goal is to separate class 2 from the other 6 classes.

**Table 4.5:** Characteristics of the binary classification UCI data sets, where  $N_{CV}$  is the number of data points used in CV based tuning procedure,  $n_{test}$  is the number of observations in the test set and  $n$  is the total data set size. The number of numerical and categorical attributes is denoted by  $d_{num}$  and  $d_{cat}$  respectively,  $d$  is the total number of attributes.

	pid	spa	mgt	adu	ftc
$n_{CV}$	512	3068	13000	33000	531012
$n_{test}$	256	1533	6020	12222	50000
$n$	768	4601	19020	45222	581012
$d_{num}$	8	57	11	6	10
$d_{cat}$	0	0	0	8	44
$d$	8	57	11	14	54

The following multi-class data sets were used: Letter Recognition (**let**), Optical recognition (**opt**), Pen based Recognition (**pen**), Statlog Landsat Satellite (**lan**) and Statlog Shuttle (**shu**) data set. The main characteristics of these data sets are summarized in Table 4.6.

**Table 4.6:** Characteristics of the multi-class classification UCI data sets. The  $M$  row denotes the number of classes for each data set encoded by  $L_{\text{MOC}}$  and  $L_{\text{1vs1}}$  bits for minimum output coding (MOC) and one-versus-one output coding (1vs1) respectively.

	opt	lan	pen	let	shu
$n_{CV}$	3750	4435	7328	13667	43500
$n_{\text{test}}$	1870	2000	3664	6333	14500
$n$	5620	6435	10992	20000	58000
$d_{\text{num}}$	64	36	16	16	9
$d_{\text{cat}}$	0	0	0	0	0
$d$	64	36	16	16	9
$M$	10	7	10	26	7
$L_{\text{MOC}}$	4	3	4	5	3
$L_{\text{1vs1}}$	45	21	45	325	21

## Regression

The following data sets for regression were also downloaded from the UCI benchmark data set: Boston Housing (**bho**) and Concrete Compressive Strength (**ccs**). The main characteristics of these data sets are given in Table 4.7.

**Table 4.7:** Characteristics of the regression UCI data sets.

	bho	ccs
$n_{CV}$	338	687
$n_{\text{test}}$	168	343
$n$	506	1030
$d_{\text{num}}$	14	9
$d_{\text{cat}}$	0	0
$d$	14	9

### 4.7.2 Description of the Reference Algorithms

The test performance of the FS-LSSVM classifier/regression model is compared to the performance of SVM and  $\nu$ -SVM (Schölkopf et al., 2000), both implemented

in the LIBSVM software. In case of  $\nu$ -SVM the parameter  $\nu \in ]0,0.8]$  is also considered as a tuning parameter. The three methods use a cache size of 1GB and the stopping criterion is set to  $10^{-3}$ . Shrinking is applied in the SVM case. For Classification, the default classifier or majority rule (Maj.Rule) is included as a baseline in the comparison tables. The majority rule (in percent) is given by the largest number of data points belonging to a class divided by total number of data points (of all classes) multiplied by hundred. All comparisons are made on the same 10 randomizations.

The comparison is performed on an out-of-sample test set consisting of 1/3 of the data. The first 2/3 of the randomized data is reserved for training and/or cross-validation. For each algorithm, the average test set performances and sample standard deviations on 10 randomizations are reported. Also the mean total time and corresponding standard deviation are mentioned. The total time of the algorithms consists of (i) 10-fold CV using the optimization strategy described in Section 3.4. The total number of function evaluations is set to 160 (90 for CSA and 70 for simplex search). In case of  $\nu$ -SVM the parameter  $\nu$  is also considered as a tuning parameter. We have used 5 multiple starters for the CSA algorithm; (ii) training with optimal tuning parameters and (iii) evaluation on test set. For FS-LSSVM we set the parameter  $k = 5$ .

### 4.7.3 Performance of binary FS-LSSVM classifiers

In what follows, the results are presented and discussed for the 7 UCI binary benchmark data sets described above. As kernel types RBF and linear (Lin) kernels were used. Performances of FS-LSSVM, SVM ( $C$ -SVC) and  $\nu$ -SVC are reported. The following experimental setup is used: each binary classifier is designed on 2/3 (random selection) of the data using 10-fold CV, while the remaining 1/3 are put aside for testing. The test set performances on the data sets are reported in Table 4.8. Table 4.9 gives the average computation time (in seconds) and standard deviation for both algorithms.

The FS-LSSVM classifier with RBF kernel (RBF FS-LSSVM) achieves the best average test performance on 3 of the 5 benchmark domains, while its accuracy is comparable to RBF SVM ( $C$ -SVC). On all binary classification data sets  $\nu$ -SVC has a slightly lower performance compared to FS-LSSVM and  $C$ -SVC. Comparison of the average test set performance achieved by the RBF kernel with the average test set performance of the linear kernel illustrates that most domains are weakly nonlinear (Holte, 1993), except for the Magic Gamma Telescope data set. Due to the high training time for SVM ( $C$ -SVC and  $\nu$ -SVC) in case of the Forest Covertype data set, it is practically impossible to perform 10-fold CV. Therefore, the values in Table 4.8 and Table 4.9 with an asterisk only denote the training time for a fixed pair of tuning parameters for SVM. No cross-validation was performed

because of the computational burden. Notice also that the FS-LSSVM models are sparser than the RBF SVM ( $C$ -SVC and  $\nu$ -SVC) models while resulting in equal or better performance on test.

**Table 4.8:** Comparison of the 10 times randomized **test set** performances (in percentage) and standard deviations (within parentheses) of FS-LSSVM (linear and RBF kernel) with the performance of  $C$ -SVC,  $\nu$ -SVC and Majority Rule classifier on 5 binary domains.  $n_{test}$  is the number of observations in the test set and  $d$  is the total number of attributes. Also the number of prototype vectors (PV) for FS-LSSVM and number of support vectors (SV) used by the algorithms are reported. The number of prototype vectors of FS-LSSVM are determined by the heuristic described in Section 4.4. The values with an asterisk only denote the performance of the  $C$ -SVC and  $\nu$ -SVC for fixed tuning parameter(s). No cross-validation was performed because of the computational burden.

	pid	spa	mgt	adu	ftc
$n_{test}$	256	1533	6020	12222	50000
$d$	8	57	11	14	54
# PV FS-LSSVM	150	200	1000	500	500
# SV $C$ -SVC	290	800	7000	11085	185000
# SV $\nu$ -SVC	331	1525	7252	12205	165205
RBF FS-LSSVM	76.7(3.43)	92.5(0.67)	86.6(0.51)	85.21(0.21)	81.8(0.52)
Lin FS-LSSVM	77.6(0.78)	90.9(0.75)	77.8(0.23)	83.9(0.17)	75.61(0.35)
RBF $C$ -SVC	75.1(3.31)	92.6(0.76)	85.6(1.46)	84.81(0.20)	81.5(*)
Lin $C$ -SVC	76.1(1.76)	91.9(0.82)	77.3(0.53)	83.5(0.28)	75.24(*)
RBF $\nu$ -SVC	75.8(3.34)	88.7(0.73)	84.2(1.42)	83.9(0.23)	81.6(*)
Maj. Rule	64.8(1.46)	60.6(0.58)	65.8(0.28)	83.4(0.1)	51.23(0.20)

**Table 4.9:** Comparison of the average computation time in seconds for the FS-LSSVM,  $C$ -SVC and  $\nu$ -SVC on 5 binary classification problems. The standard deviation is shown within parentheses. The values with an asterisk only denotes the training time for a fixed pair of tuning parameters for  $C$ -SVC and  $\nu$ -SVC. No cross-validation was performed because of the computational burden.

Av. Time (s)	pid	spa	mgt	adu	ftc
RBF FS-LSSVM	30.1(1.9)	249(16)	9985(112)	7344(295)	122290(989)
Lin FS-LSSVM	19.8(0.5)	72(3.8)	1298(13)	1404(47)	5615(72)
RBF $C$ -SVC	24.8(3.1)	1010(53)	20603(396)	139730(5556)	58962(*)
Lin $C$ -SVC	18(0.65)	785(22)	13901(189)	130590(4771)	53478(*)
RBF $\nu$ -SVC	30.3(2.3)	1372(43)	35299(357)	139927(3578)	55178(*)

#### 4.7.4 Performance of multi-class FS-LSSVM classifiers

Each multi-class problem is decomposed in a set of binary classification problems using minimum output coding (MOC) and one-versus-one (1vs1) output coding for FS-LSSVM and one-versus-one (1vs1) output coding for SVM ( $C$ -SVC and  $\nu$ -SVC). The same kernel types as in the binary classification problem are considered. Performances of FS-LSSVM and SVM ( $C$ -SVC and  $\nu$ -SVC) are reported. We used the same experimental setup as for binary classification. The test set performances on the different data sets are reported in Table 4.10. Table 4.11 gives the average computation time (in seconds) and standard deviation for the three algorithms. Performance on test as well as the accuracy of the multi-class FS-LSSVM and multi-class SVM ( $C$ -SVC and  $\nu$ -SVC) are similar. From Table 4.10 it is clear that there is a difference between the encoding schemes. In general, 1vs1 output coding results in better performances on test than minimum output coding (MOC). This can be especially seen from the Lin FS-LSSVM result on the Letter Recognition data set. Notice that the FS-LSSVM models are again sparser than the two types of SVM models.

**Table 4.10:** Comparison of the 10 times randomized **test set** performances (in percentage) and standard deviations (within parentheses) of FS-LSSVM (RBF kernel) with the performance of  $C$ -SVC,  $\nu$ -SVC and Majority Rule classifier on 5 multi-class domains using MOC and 1vs1 output coding.  $n_{\text{test}}$  is the number of observations in the test set and  $d$  is the total number of attributes. Also the number of prototype vectors (PV) and number of support vectors (SV) used by the algorithms are reported. The number of prototype vectors of FS-LSSVM are determined by the heuristic described in Section 4.4.

	opt	lan	pen	let	shu
$n_{\text{test}}$	1870	2000	3664	6333	14500
$d$	64	36	16	16	9
# PV FS-LSSVM	420	330	250	1500	175
# SV $C$ -SVC	3750	1876	1178	8830	559
# SV $\nu$ -SVC	2810	2518	4051	11359	521
RBF FS-LSSVM (MOC)	96.87(0.70)	91.83(0.43)	99.44(0.17)	89.14(0.22)	99.87(0.03)
RBF FS-LSSVM (1vs1)	98.14(0.10)	91.93(0.3)	99.57(0.10)	95.65(0.17)	99.84(0.03)
Lin FS-LSSVM (1vs1)	97.18(0.35)	85.71(0.77)	96.67(0.35)	84.87(0.49)	96.82(0.18)
Lin FS-LSSVM (MOC)	78.62(0.32)	74.35(0.26)	68.21(0.36)	18.20(0.46)	84.78(0.33)
RBF $C$ -SVC (1vs1)	97.73(0.14)	92.14(0.45)	99.51(0.13)	95.67(0.19)	99.86(0.03)
Lin $C$ -SVC (1vs1)	97.21(0.26)	86.12(0.79)	97.52(0.62)	84.96(0.56)	97.02(0.19)
RBF $\nu$ -SVC (1vs1)	95.3(0.12)	88.3(0.31)	95.96(0.16)	93.18(0.21)	99.34(0.03)
Maj. Rule	10.45(0.12)	23.61(0.16)	10.53(0.07)	4.10(0.14)	78.81(0.04)



**Table 4.11:** Comparison of the average computation time in seconds for the FS-LSSVM, C-SVM and  $\nu$ -SVC on 5 multi-class classification problems. The standard deviation is shown within parentheses.

Av. Time (s)	opt	lan	pen	let	shu
RBF FS-LSSVM (MOC)	4892(162)	2159(83)	2221(110)	105930(2132)	5908(272)
RBF FS-LSSVM (1vs1)	623(36)	739(17)	514(42)	10380(897)	2734(82)
Lin FS-LSSVM (1vs1)	282(19)	153(6)	156(4)	2792(11)	501(8)
Lin FS-LSSVM (MOC)	942(6)	409(13)	279(10)	44457(1503)	645(31)
RBF C-SVC (1vs1)	11371(573)	6612(347)	11215(520)	59102(2412)	52724(3619)
Lin C-SVC (1vs1)	474(1)	1739(48)	880(16)	11203(467)	50174(2954)
RBF $\nu$ -SVC (1vs1)	7963(178)	8229(304)	16589(453)	79040(2354)	50478(2879)

#### 4.7.5 Performance of FS-LSSVM for Regression

We have used the same preprocessing and tuning as in the classification case. The test performances of the data sets are given in Table 4.12. Table 4.13 reports the average computation time (in seconds) and standard deviations for both algorithms. In each of the regression examples the RBF kernel is used. From these results it can be seen that our algorithm has better performances and smaller standard deviations than  $\varepsilon$ -SVR and  $\nu$ -SVR. FS-LSSVM results into a sparser model for both data sets compared to  $\varepsilon$ -SVR.

**Table 4.12:** Comparison of the 10 times randomized **test set** performances ( $L_2$ ,  $L_1$ ,  $L_\infty$ ) and standard deviations (within parentheses) of FS-LSSVM (RBF kernel) on 2 regression domains.

		bho	ccs
$n_{\text{test}}$		168	343
$d$		14	9
# PV FS-LSSVM		200	120
# SV $\varepsilon$ -SVR		226	670
# SV $\nu$ -SVR		195	330
RBF FS-LSSVM	$L_2$	0.13(0.02)	0.17(0.02)
	$L_1$	0.24(0.02)	0.30(0.03)
	$L_\infty$	1.90(0.50)	1.22(0.42)
RBF $\varepsilon$ -SVR	$L_2$	0.16(0.05)	0.23(0.02)
	$L_1$	0.24(0.03)	0.33(0.02)
	$L_\infty$	2.20(0.54)	1.63(0.58)
RBF $\nu$ -SVR	$L_2$	0.16(0.04)	0.22(0.02)
	$L_1$	0.26(0.03)	0.34(0.03)
	$L_\infty$	1.97(0.58)	1.72(0.52)

**Table 4.13:** Comparison of the average computation time in seconds for the FS-LSSVM,  $\varepsilon$ -SVR and  $\nu$ -SVR on 2 regression problems. The standard deviation is shown within parentheses.

Av. Time (s)	bho	ccs
RBF FS-LSSVM	74(2)	94(3)
RBF $\varepsilon$ -SVR	63(1)	168(3)
RBF $\nu$ -SVR	61(1)	131(2)

## 4.8 Conclusions

In this Chapter, we elucidated the problem with kernel based methods when considering large data sets. For LS-SVM, we estimated a finite  $m$ -approximate feature map based on the Nyström approximation so that the problem could be solved in the primal space. In order to select proper prototype vectors, we used the quadratic Rényi entropy. Also, we have illustrated how to select the bandwidth for the entropy estimation in a fast and reliable way using the solve-the-equation plug-in method. Further, we have shown that this entropy criterion with no additional moment constraints is maximized by a uniform density over the input space. In order to select the tuning parameters for large scale data sets, a fast cross-validation procedure was developed. Finally, the performance of FS-LSSVM is compared to different methods on several data sets. The speed-up achieved by our algorithm is about 10 to 20 times compared to LIBSVM. We observed that our method requires less prototype vectors than support vectors in SVM, hence resulting into sparser models.

## Chapter 5

# Robustness in Kernel Based Regression

In the previous Chapters, basic methods for LS-SVM and FS-LSSVM regression were discussed. The use of an  $L_2$  loss function and equality constraints for the models results into simpler formulations but on the other hand they have a potential drawback such as the lack of robustness. In this Chapter we will robustify LS-SVM and FS-LSSVM via iteratively reweighting. In order to understand the robustness of these estimators against outliers, we use the empirical influence function and empirical maxbias curves. Contributions are made in Section 5.4.

### 5.1 Introduction

Regression analysis is an important statistical tool routinely applied in most sciences. However, using least squares techniques, there is an awareness of the dangers posed by the occurrence of outliers present in the data. Not only the response variable can be outlying, but also the explanatory part, leading to leverage points. Both types of outliers may totally spoil an ordinary LS analysis.

To cope with this problem, statistical techniques have been developed that are not so easily affected by outliers. These methods are called robust or resistant. A *first attempt* was done by Edgeworth (Edgeworth, 1887). He argued that outliers have a very large influence on LS because the residuals are squared. Therefore, he proposed the least absolute values regression estimator ( $L_1$  regression).

The *second great step* forward in this class of methods occurred in the 1960s and early 1970s with fundamental work of Tukey (Tukey, 1960), Huber (Huber,

1964) (minimax approach) and Hampel (influence functions) (Hampel, 1971). Huber (Huber, 1964) gave the first theory of robustness. He considered the general gross-error model or  $\epsilon$ -contamination model

$$\mathcal{G}_\epsilon = \{F : F(x) = (1 - \epsilon)F_0(x) + \epsilon G(x), 0 \leq \epsilon \leq 1\}, \quad (5.1)$$

where  $F_0$  is some given distribution (the ideal nominal model),  $G$  is an arbitrary continuous distribution and  $\epsilon$  is the first parameter of contamination. This contamination model describes the case, where with large probability  $(1 - \epsilon)$ , the data occurs with distribution  $F_0$  and with small probability  $\epsilon$  outliers occur according to distribution  $G$ .

**Example 5.1**  *$\epsilon$ -contamination model with symmetric contamination*

$$F(x) = (1 - \epsilon)\mathcal{N}(0,1) + \epsilon\mathcal{N}(0,\kappa^2\sigma^2), \quad 0 \leq \epsilon \leq 1, \kappa > 1.$$

**Example 5.2**  *$\epsilon$ -contamination model for the mixture of the Normal and Laplace or double exponential distribution*

$$F(x) = (1 - \epsilon)\mathcal{N}(0,1) + \epsilon\text{Lap}(0,\lambda), \quad 0 \leq \epsilon \leq 1, \lambda > 0.$$

Huber considered also the class of  $M$ -estimators of location (also called generalized maximum likelihood estimators) described by some suitable function. The Huber estimator is a minimax solution: it minimizes the maximum asymptotic variance over all  $F$  in the gross-error model.

Huber developed a second theory (Huber, 1965, 1968; Huber and Strassen, 1973, 1974) for censored likelihood ratio tests and exact finite sample confidence intervals, using more general neighborhoods of the normal model. This approach may be mathematically the most rigorous but seems very hard to generalize and therefore plays hardly any role in applications. A third theory proposed by Hampel (Hampel, 1968, 1971, 1974; Hampel et al., 1986) is closely related to robustness theory which is more generally applicable than Huber's first and second theory. Three main concepts are introduced: (i) qualitative robustness, which is essentially continuity of the estimator viewed as functional in the weak topology; (ii) the Influence Curve (IC) or Influence Function (IF), which describes the first derivative of the estimator, as far as existing; and (iii) the Breakdown Point (BP), a global robustness measure describing how many percent gross errors are still tolerated before the estimator totally breaks down.

Robustness has provided at least two major insights into statistical theory and practice: (i) Relatively small perturbations from nominal models can have very substantial deleterious effects on many commonly used statistical procedures and methods (e.g. estimating the mean, F-test for variances). (ii) Robust methods

are needed for detecting or accommodating outliers in the data (Hubert, 2001; Debruyne et al., 2009; Debruyne, 2009).

From their work the following methods were developed:  $M$ -estimators, Generalized  $M$ -estimators,  $R$ -estimators,  $L$ -estimators,  $S$ -estimators, repeated median estimator, least median of squares, . . . Detailed information about these estimators as well as methods for robustness measuring can be found in the books by Hampel et al. (1986), Rousseeuw and Leroy (2003), Maronna et al. (2006) and Huber and Ronchetti (2009). See also the book by Jurečková and Pícek (2006) for robust statistical methods with R (a language and environment for statistical computing and graphics freely available at <http://cran.r-project.org/bin/windows/base/>) providing a systematic treatment of robust procedures with an emphasis on practical applications.

## 5.2 Measures of Robustness

In order to understand why certain estimators behave the way they do, it is necessary to look at various measures of robustness. There exist numerous approaches towards the robustness problem. The approach based on influence functions will be used here. The effect of one outlier on the estimator can be described by the influence function (IF). The IF describes the (approximate and standardized) effect of an additional observation in any point  $x$  on a statistic  $T$ , given a (large) sample with distribution  $F$ . Another measure of robustness of an estimator is the maxbias curve. The maxbias curve gives the maximal bias that an estimator can suffer from when a fraction of the data come from a contaminated distribution. By letting the fraction vary between zero and the breakdown value a curve is obtained. The breakdown value is defined as how much contaminated data an estimator can tolerate before it becomes useless.

### 5.2.1 Influence Functions and Breakdown Points

Let  $F$  be a fixed distribution and  $T(F)$  a statistical functional defined on a set  $\mathcal{G}_\epsilon$  of distributions satisfying that  $T$  is Gâteaux differentiable at the distribution  $F$  in  $\text{domain}(T)$  (Hampel et al., 1986). Let the estimator  $T(\hat{F}_n)$  of  $T(F)$  be the functional of the sample distribution  $F_n$ .

**Definition 5.1 (Influence Function)** *The influence function (IF) of  $T$  at  $F$  is given by*

$$\text{IF}(x; T, F) = \lim_{\epsilon \downarrow 0} \frac{T[(1 - \epsilon)F + \epsilon\Delta_x] - T(F)}{\epsilon} \quad (5.2)$$

in those  $x$  where this limit exists.  $\Delta_x$  denotes the probability measure which puts mass 1 at the point  $x$ .

Hence, the IF reflects the bias caused by adding a few outliers at the point  $x$ , standardized by the amount  $\epsilon$  of contamination. Therefore, a bounded IF leads to robust estimators. Note that this kind of differentiation of statistical functionals is a differentiation in the sense of von Mises with a kernel function (Fernholz, 1983; Clark, 1983). From the influence function, several robustness measures can be defined: the gross error sensitivity, the local shift sensitivity and the rejection point, see Hampel et al. (1986, Section 2.1c) for an overview. Mathematically speaking, the influence function is the set of all partial derivatives of the functional  $T$  in the direction of the point masses. For functionals, there exist several concepts of differentiation i.e. Gâteaux, Hadamard or compact, Bouligand and Fréchet. An application of the Bouligand IF can be found in Christmann and Messem (2008) in order to investigate the robustness properties of SVMs. The Bouligand IF has the advantage of being positive homogeneous which is in general not true for Hampel's influence function (5.2). Christmann and Messem (2008) also show that there exists an interesting relationship between the Bouligand IF and the IF: if the Bouligand IF exists, then the IF does also exist and both are equal.

Next, we give the definitions of the maxbias curve and the breakdown point. Note that some authors can give a slightly different definition of the maxbias curve, see e.g. Croux and Haesbroeck (2001).

**Definition 5.2 (Maxbias Curve)** Let  $T(F)$  denote a statistical functional and let the contamination neighborhood of  $F$  be defined by  $\mathcal{G}_\epsilon$  for a fraction of contamination  $\epsilon$ . The maxbias curve is defined by

$$B(\epsilon, T, F) = \sup_{F \in \mathcal{G}_\epsilon} |T(F) - T(F_0)|. \quad (5.3)$$

**Definition 5.3 (Breakdown Point)** The breakdown point  $\epsilon^*$  of an estimator  $T(\hat{F}_n)$  for the functional  $T(F)$  at  $F$  is defined by

$$\epsilon^*(T, F) = \inf\{\epsilon > 0 | B(\epsilon, T, F) = \infty\}.$$

From the previous definition it is obvious that the breakdown point defines the largest fraction of gross errors that still keeps the bias bounded. We will give some examples of influence functions and breakdown points for the mean, median and variance.

**Example 5.3 (Mean)** The corresponding functional  $T(F) = \int x dF(x)$  of the mean is defined for all probability measures with existing first moment. From (5.2),

it follows that

$$\begin{aligned}
 \text{IF}(x; T, F) &= \lim_{\epsilon \downarrow 0} \frac{\int x d[(1 - \epsilon)F + \epsilon\Delta_x](x) - \int x dF(x)}{\epsilon} \\
 &= \lim_{\epsilon \downarrow 0} \frac{(1 - \epsilon) \int x dF(x) + \epsilon \int x d\Delta_x(x) - \int x dF(x)}{\epsilon} \\
 &= \lim_{\epsilon \downarrow 0} \frac{\epsilon \int x dF(x) + \epsilon \int x d\Delta_x(x)}{\epsilon} \\
 &= \lim_{\epsilon \downarrow 0} \frac{\epsilon x - \epsilon T(F)}{\epsilon} \\
 &= x - T(F).
 \end{aligned}$$

Hence, the IF of the sample mean is clearly unbounded in  $\mathbb{R}$ , see Figure 5.1. This means that an added observation at a large distance from  $T(F)$  gives a large value in absolute sense for the IF. The finite sample breakdown point of the sample mean is  $\epsilon^* = 1/n$  but often the limiting value  $\lim_{n \rightarrow \infty} 1/n = 0$  is used as a measure of the global stability of the estimator. One of the more robust location estimators is the median.

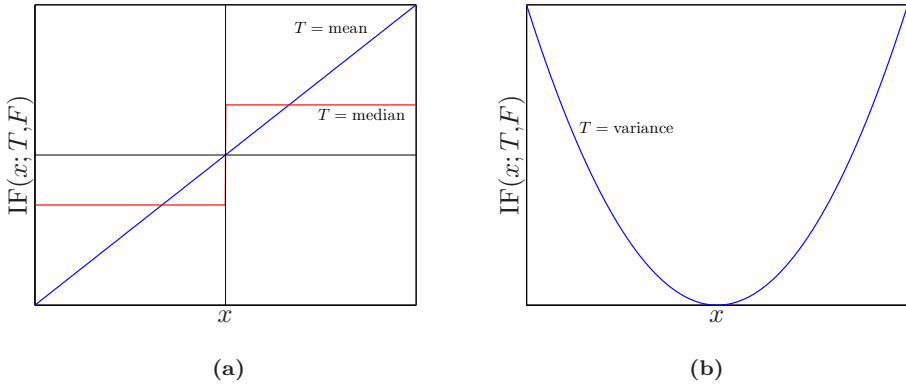
**Example 5.4 (Median)** *The corresponding functional  $T(F) = F^{-1}(\frac{1}{2})$  of the median is defined for all probability measures. Using the influence function of an  $M$ -estimator yields*

$$\begin{aligned}
 \text{IF}(x; T, F) &= \frac{\text{sign}(x)}{[\text{sign}(p+) - \text{sign}(p-)]f(p)} \\
 &= \frac{\text{sign}(x)}{2f(0)},
 \end{aligned}$$

where  $p$  is the point of discontinuity with left and right limits  $\text{sign}(p+) \neq \text{sign}(p-)$  and  $f$  is the density.

Because the IF of the median has a jump at zero (Figure 5.1a), the median is sensitive to wiggling near the center of symmetry. A surprising fact is that the median does not reject outliers. Indeed, if one adds an outlier to the sample, then the middle order statistic will move in that direction. The breakdown point of the median is 0.5 and its asymptotic efficiency is low.

**Example 5.5 (Variance)** *The corresponding functional  $T(F) = \int (x - \mathbf{E}[x])^2 dF(x)$  of the variance is defined for all probability measures with existing first and second*



**Figure 5.1:** (a) Influence functions of the mean and median. The influence function of the mean is unbounded in  $\mathbb{R}$  while the influence function of the median is bounded in  $\mathbb{R}$ ; (b) Influence function of the variance. The influence function of the variance is unbounded in  $\mathbb{R}$ .

moments. From (5.2), it follows that

$$\begin{aligned}
 \text{IF}(x; T, F) &= \lim_{\epsilon \downarrow 0} \frac{\int (x - \mathbf{E}[x])^2 d[(1 - \epsilon)F + \epsilon\Delta_x](x) - \int (x - \mathbf{E}[x])^2 dF(x)}{\epsilon} \\
 &= \lim_{\epsilon \downarrow 0} \frac{\epsilon \int (x - \mathbf{E}[x])^2 d\Delta_x(x) - \epsilon \int (x - \mathbf{E}[x])^2 dF(x)}{\epsilon} \\
 &= (x - \mathbf{E}[X])^2 - T(F).
 \end{aligned}$$

The influence function of the variance is shown in Figure 5.1b and is unbounded in  $\mathbb{R}$ . This means that an added observation at a large distance from  $T(F)$  gives a large value in absolute sense for the IF. The finite sample breakdown point of the sample variance is  $\epsilon^* = 1/n$  but often the limiting value  $\lim_{n \rightarrow \infty} 1/n = 0$  is used as a measure of the global stability of the estimator.

## 5.2.2 Empirical Influence Functions

The definition of the influence function (5.2) is entirely asymptotic because it is focused on functionals which coincide with the estimator's asymptotic value. However, there exist also some simple finite-sample or empirical versions, which can be easily computed. The most important empirical influence functions are the sensitivity curve (Tukey, 1977) and the Jackknife (Quenouille, 1956; Tukey, 1958).



### Sensitivity Curve

There exist two version of the sensitivity curve i.e. one with addition and one with replacement. In case of an additional observation, one starts with a sample  $(x_1, \dots, x_{n-1})$  of size  $n - 1$ . Let  $T_n(\hat{F}_{n-1}) = T_n(x_1, \dots, x_{n-1})$  be the estimator of  $T(F_{n-1})$ . The change in estimate when an  $n^{\text{th}}$  observation  $x_n = x$  is included gives  $T_n(x_1, \dots, x_{n-1}, x) - T_n(x_1, \dots, x_{n-1})$ . Multiplying this change by  $n$  results in the sensitivity curve

**Definition 5.4 (Sensitivity Curve)** *The sensitivity curve is obtained by replacing  $F$  by  $\hat{F}_{n-1}$  and  $\epsilon$  by  $\frac{1}{n}$  in (5.2):*

$$\begin{aligned} \text{SC}_n(x; T, \hat{F}_{n-1}) &= \frac{T\left[\left(\frac{n-1}{n}\right)\hat{F}_{n-1} + \frac{1}{n}\Delta_x\right] - T(\hat{F}_{n-1})}{\frac{1}{n}} & (5.4) \\ &= (n-1)T(\hat{F}_{n-1}) + T(\Delta_x) - nT(\hat{F}_{n-1}) \\ &= n[T_n(x_1, \dots, x_{n-1}, x) - T_{n-1}(x_1, \dots, x_{n-1})]. \end{aligned}$$

It can be seen that (5.4) is a special case of (5.2), with  $F_{n-1}$  as an approximation for  $F$  and with contamination size  $\epsilon = \frac{1}{n}$ . In many situations,  $\text{SC}_n(x; T, \hat{F}_{n-1})$  will therefore converge to  $\text{IF}(x; T, F)$  when  $n \rightarrow \infty$ .

**Example 5.6 (Mean)** *Let  $T(F) = \mu = \mathbf{E}[X]$  denote the mean in a population and let  $x_1, \dots, x_{n-1}$  denote a sample from that population. The sensitivity curve of the mean is given by*

$$\begin{aligned} \text{SC}_n(x; \mu, \hat{F}_{n-1}) &= n[\hat{\mu}(x_1, \dots, x_{n-1}, x) - \hat{\mu}(x_1, \dots, x_{n-1})] \\ &= (n-1)\hat{\mu}(x_1, \dots, x_{n-1}) - n\hat{\mu}(x_1, \dots, x_{n-1}) + x \\ &= x - \hat{\mu}(x_1, \dots, x_{n-1}). \end{aligned}$$

**Example 5.7 (Median)** *The sample median is defined as*

$$\text{med} = \begin{cases} x_{n(k+1)}, & \text{if } n = 2k + 1; \\ \frac{x_{n(k)} + x_{n(k+1)}}{2}, & \text{if } n = 2k, \end{cases}$$

where  $x_{n(1)} \leq \dots \leq x_{n(n)}$  are the order statistics. The sensitivity curve of the median is given by

$$\text{SC}_n(x; \text{med}, \hat{F}_{n-1}) = \begin{cases} n[x_{n(k)} - \text{med}(x_1, \dots, x_{n-1})], & \text{if } x < x_{n(k)}; \\ x, & \text{if } x_{n(k)} \leq x \leq x_{n(k+1)}; \\ n[x_{n(k+1)} - \text{med}(x_1, \dots, x_{n-1})], & \text{if } x > x_{n(k+1)}. \end{cases}$$

Given a univariate data set with  $X \sim \mathcal{N}(0,1)$ . The following location estimators are applied to the sample: sample mean and sample median. The sensitivity curve for both location estimators is shown in Figure 5.2a. The sensitivity curve for the mean becomes unbounded for both  $X \rightarrow \infty$  and  $X \rightarrow -\infty$ , whereas the sensitivity curve of the median remains bounded.

**Example 5.8 (Variance)** Let  $T(F) = \sigma^2$  denote the variance of a population and let  $x_1, \dots, x_n$  denote a sample from that population. An estimate of the variance is given by  $\hat{\sigma}_n^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \hat{\mu}_n)^2$  where  $\hat{\mu}_n$  denotes the sample mean computed over  $n$  data points. Also shift the horizontal axis so that  $\sum_{i=1}^{n-1} x_i = 0$  i.e.  $\hat{\mu}_{n-1} = 0$ . The sensitivity curve of the sample variance is given by

$$\begin{aligned} \text{SC}_n(x; \hat{\sigma}^2, \hat{F}_{n-1}) &= n[\hat{\sigma}_n^2 - \hat{\sigma}_{n-1}^2] \\ &= \sum_{i=1}^n x_i^2 - n\hat{\mu}_n^2 - n\hat{\sigma}_{n-1}^2 \\ &= \sum_{i=1}^{n-1} x_i^2 + x^2 - \frac{x^2}{n} - n\hat{\sigma}_{n-1}^2 \\ &= \sum_{i=1}^{n-1} x_i^2 - (n-1)\hat{\mu}_{n-1}^2 + \frac{n-1}{n}x^2 - n\hat{\sigma}_{n-1}^2 \\ &= \frac{n-1}{n}x^2 - \hat{\sigma}_{n-1}^2. \end{aligned}$$

For  $n \rightarrow \infty$ ,  $\text{SC}_n(x; \hat{\sigma}^2, \hat{F}_{n-1})$  converges to  $\text{IF}(x; \sigma^2, F)$ .

Given a univariate data set with  $X \sim \mathcal{N}(0,1)$ . The following scale estimators are applied to the sample: variance, mean absolute deviation and Median Absolute Deviation (MAD). The mean absolute deviation is defined as

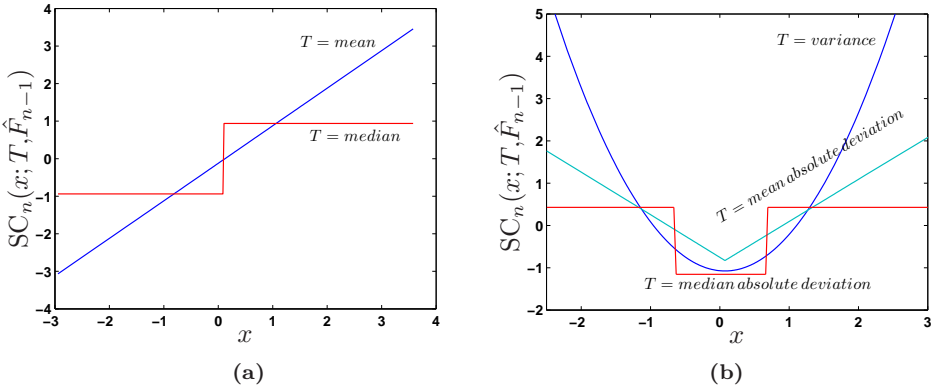
$$T(\hat{F}_n) = \frac{1}{n} \sum_{i=1}^n \left| X_i - \frac{1}{n} \sum_{k=1}^n X_k \right|.$$

This estimator is nonrobust to outliers and has a breakdown point  $\epsilon^* = 0$ . The MAD, a more robust scale estimator, is defined as

$$T(\hat{F}_n) = \text{med}_i(|X_i - \text{med}(X_1, \dots, X_n)|).$$

This estimator is robust to outliers and has a breakdown point  $\epsilon^* = 0.5$ . The sensitivity curves for the three scale estimators are shown in Figure 5.2b. The variance and the mean absolute deviation become unbounded for both  $X \rightarrow \infty$

and  $X \rightarrow -\infty$ , whereas the sensitivity curve of the MAD remains bounded. Better robust estimates of scale are the  $S_n$  and  $Q_n$  estimator proposed by Rousseeuw and Croux (1993). These estimators have a breakdown point  $\epsilon^* = 0.5$  and have a better efficiency than MAD for Gaussian models as well as for non Gaussian models.



**Figure 5.2:** (a) Empirical IF (sensitivity curve) of the mean and median. The influence function of the mean is unbounded in  $\mathbb{R}$  while the influence function of the median is bounded in  $\mathbb{R}$ ; (b) Empirical influence function of the variance, mean absolute deviation and median absolute deviation. The empirical IF of the variance and mean absolute deviation is unbounded in  $\mathbb{R}$  whereas the empirical IF median absolute deviation is bounded.

**Jackknife Approximation**

An other approach to approximate the IF, but only at the sample values  $x_1, \dots, x_n$  is the Jackknife.

**Definition 5.5 (Jackknife Approximation)** *Substituting  $\hat{F}_n$  for  $F$  and setting  $\epsilon = \frac{-1}{n-1}$  in (5.2), one obtains*

$$\begin{aligned} \text{IF}_{JA}(x_i; T, \hat{F}_n) &= \frac{T \left[ \left( \frac{n}{n-1} \right) \hat{F}_n - \frac{1}{n-1} \Delta_{x_i} \right] - T(\hat{F}_n)}{-\frac{1}{n-1}} \\ &= (n-1)[T_n(x_1, \dots, x_n) - T_{n-1}(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n)]. \end{aligned}$$

In some cases, namely when the IF does not depend smoothly on  $F$ , the Jackknife is in trouble.

## 5.3 Residuals and Outliers in Regression

Residuals are used in many procedures designed to detect various types of disagreement between the data and an assumed model. In this Section, we consider observations that do not belong to the model and often exhibit numerically large residuals. In this case these observations are called outliers. Although the detection of outliers in a univariate sample has been investigated extensively in the statistical literature (e.g. Barnett and Lewis (1984)), the word outlier has never been given a precise definition. In this thesis we use the one of Barnett and Lewis (1984). A quantitative definition has been given by Davis and Gather (1993).

**Definition 5.6 (Barnett and Lewis, 1984)** *An outlier is an observation (or subset of observations) which appears to be inconsistent with the remainder of that set of data.*

An introduction to residuals and outliers is given by Fox (1991). More advanced treatments are given by Cook and Weisberg (1982) and by Atkinson and Riani (2000).

### 5.3.1 Linear Regression

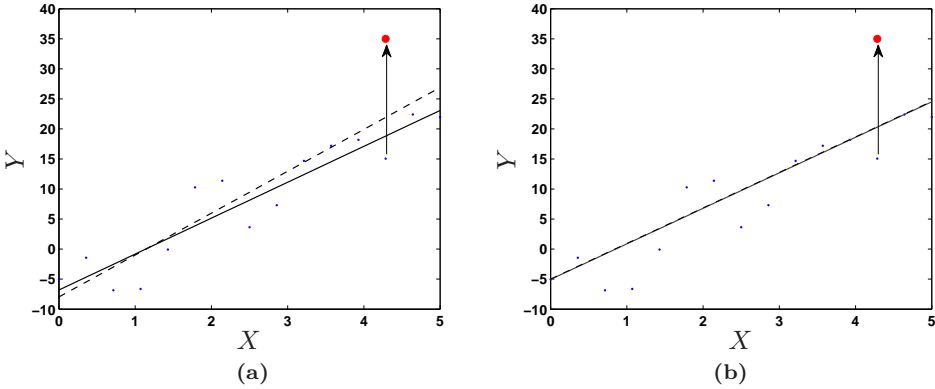
A simple linear regression model assumes the following relation

$$Y_k = \beta_1 X_k + \beta_0 + e_k, \quad k = 1, \dots, n,$$

in which the slope  $\beta_1$  and the intercept  $\beta_0$  have to be estimated from data. The slope and intercept are often determined by the Least Squares (LS) principle with loss function  $\mathcal{L}$  i.e. the parameters  $\beta_1$  and  $\beta_0$  are found by minimizing

$$(\hat{\beta}_0, \hat{\beta}_1) = \arg \min_{\beta_0, \beta_1 \in \mathbb{R}^2} \left\{ \frac{1}{n} \sum_{k=1}^n \mathcal{L}(Y_k, m(X_k)) \right\}$$

with  $m(X_k) = \beta_1 X_k + \beta_0$ . It is well-known that by taking a squared loss, an outlier has a large influence on the LS regression line. On the other hand, taking an  $L_1$  loss leads to robust estimates w.r.t. outliers in the  $Y$ -direction. Figure 5.3 illustrates the effect of an outlier on  $L_2$  and  $L_1$  regression. Although  $L_1$  regression leads to robust estimates in case of outliers in the  $Y$ -direction, it certainly does not have the same property for an outlier in the  $X$ -direction (leverage point). In order to obtain a fully robust method, Rousseeuw and Leroy (2003) developed the least median of squares and least trimmed squares estimators. Also the use of decreasing kernels, i.e. kernels such that  $K(u) \rightarrow 0$  when  $u \rightarrow \infty$ , leads to quite robust methods w.r.t. to leverage points. The influence for both  $x \rightarrow \infty$  and  $x \rightarrow -\infty$  is bounded



**Figure 5.3:** The original data and one outlier in the  $Y$ -direction. (a) The solid line corresponds to the  $L_2$  regression estimate without the outlier. The dashed line corresponds to  $L_2$  regression estimate with the outlier. The influence of the outlier is clearly visible in the figure; (b) The solid line corresponds to the  $L_1$  regression estimate without the outlier. The dashed line corresponds to  $L_1$  regression estimate with the outlier. The  $L_1$  regression estimate is not affected by the outlier.

in  $\mathbb{R}$  when using decreasing kernels. Common choices for decreasing kernels are:  $K(u) = \max(1 - u^2, 0)$ ,  $K(u) = \exp(-u^2)$  and  $K(u) = \exp(-u)$ .

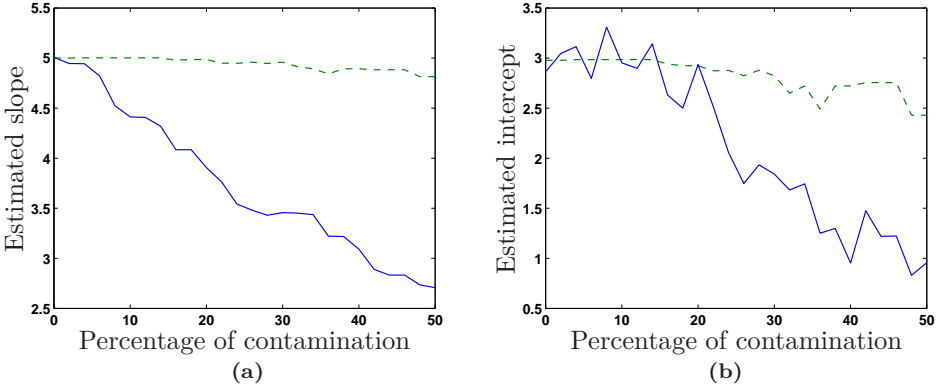
In the following toy example we will illustrate how the  $L_1$  and  $L_2$  regression estimators deal with several outliers in the data set. Given 50 “good” observations according to the linear relation

$$Y_k = 5X_k + 3 + e_k, \quad k = 1, \dots, 50,$$

where  $e \sim \mathcal{N}(0, 0.5^2)$  and  $X \sim \mathcal{U}(0, 5)$ . Applying the  $L_1$  and  $L_2$  regression estimators to this data yield values of the slope and intercept which are close to the original values. In what follows we will contaminate the data by deleting, at each step, a “good” point and replace it with a “bad” point  $Y_k^b$ , generated according to  $Y_k^b \sim \mathcal{N}(2, 6^2)$ . This was repeated until 25 “good” points remained. Figure 5.4 illustrates the breakdown plot where the value of the slope and the intercept are given as a function of the contamination percentage. It is clearly visible that the  $L_2$  estimates are immediately affected by the outliers, whereas the  $L_1$  estimates almost do not change.

A disadvantage of robust methods is their lack of efficiency when the errors are normally distributed. In order to improve the efficiency of these robust methods reweighted least squares is often used. First, one calculates the residuals obtained by a robust estimator. Then, the residuals are given a weight according to their

value and chosen weight function leading to weighted observations. Finally, a standard LS method can be used to obtain the final estimate.



**Figure 5.4:** Breakdown plot. Solid line corresponds to  $L_2$  regression and dashed line to  $L_1$  regression. (a) Estimated slope as a function of the contamination percentage; (b) Estimated intercept as a function of the contamination percentage.

### 5.3.2 Kernel Based Regression

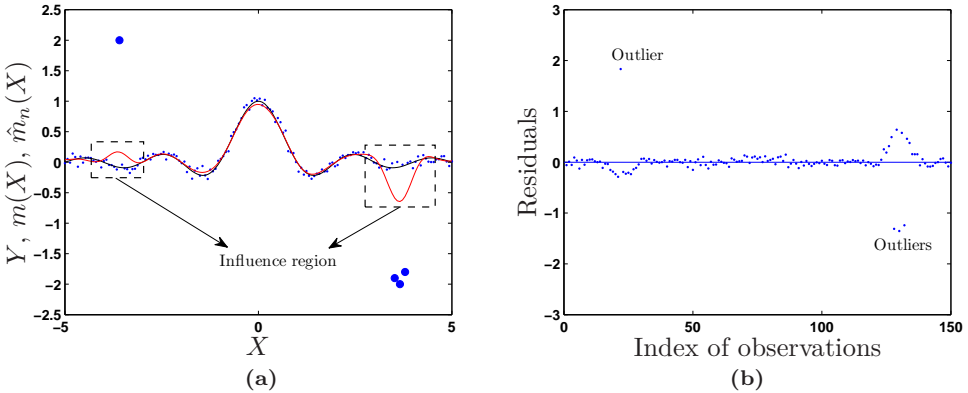
Recall that the one dimensional LS-SVM regression estimate  $\hat{m}_n$  (see Chapter 2) is given by

$$\hat{m}_n(x) = \sum_{k=1}^n \hat{\alpha}_k K\left(\frac{x - X_k}{h}\right) + \hat{b},$$

where  $\hat{\alpha}_k \in \mathbb{R}$  and  $\hat{b} \in \mathbb{R}$ . Since the primal LS-SVM formulation is based on a squared loss, the estimate cannot be expected to be robust against outliers in the  $Y$ -direction. Although the estimate is not robust, one does not observe a similar behavior as in the parametric case i.e. global breakdown of the estimate as a result of even one single outlier. In case of LS-SVM (also for NW, local polynomial regression, etc.), one observes only a local effect on the estimate (influence region) due to the outlier or groups of outliers. This behavior is illustrated in Figure 5.5a for a single outlier and a group of outliers in the  $Y$ -direction. Hence, residuals obtained from a robust estimator embody powerful information to detect outliers present in the data. This behavior is of course only observed when a small number of outliers are present in the data set.

Let  $(X_i, Y_i^b)$  be an outlier ( $Y$ -direction) and let  $\mathcal{A}$  be the influence region. In general (in case of kernel based regression), an outlier will have a relatively

larger influence on the estimate  $\hat{m}_n(X_i)$  when  $(X_i, \hat{m}_n(X_i)) \in \mathcal{A}$  than when  $(X_j, \hat{m}_n(X_j)) \notin \mathcal{A}$ . Also, the residuals from kernel based regression estimates are very useful as outlier diagnostics. Figure 5.5b gives evidence of the presence of an outlying observation and a group of outliers.



**Figure 5.5:** (a) The effects of a single outlier and a group of outliers in the Y-direction on LS-SVM regression. The estimate is only effected in a neighborhood of the outliers (influence region); (b) Residual plot associated with LS-SVM regression. From this plot we can conclude that the data set contains one outlier and a group of outliers.

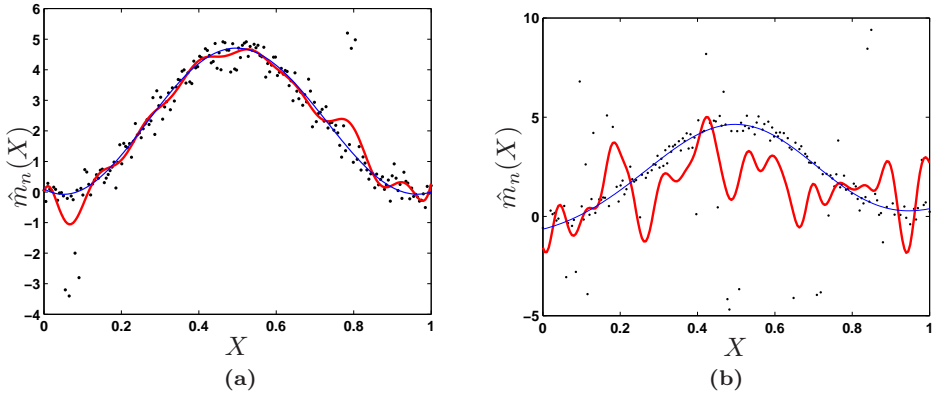
## 5.4 Robustifying LS Kernel Based Regression

In the previous Section we have illustrated that taking a non-robust loss function, e.g.  $L_2$ , can totally spoil the LS estimate in the presence of only one outlying observation. In case of Kernel Based Regression (KBR) based on an  $L_2$  loss, the estimate is only affected in an influence region if the number of outliers is small. In this Section we will demonstrate that even if the initial estimate is non-robust, we can obtain a robust estimate via iteratively reweighting. However, there is another important issue influencing the KBR estimate when outliers are present in the data i.e. the model selection. Before summarizing some theoretical results, we will demonstrate how model selection criteria can influence the final result.

### 5.4.1 Problems with Outliers in Nonparametric Regression

Consider 200 observations on the interval  $[0,1]$  and a low-order polynomial mean function  $m(X) = 300(X^3 - 3X^4 + 3X^5 - X^6)$ . Figure 5.6a shows the mean function

with normally distributed errors with variance  $\sigma^2 = 0.3^2$  and two distinct groups of outliers. Figure 5.6b shows the same mean function, but the errors are generated from the gross error or  $\epsilon$ -contamination model (5.1). In this simulation  $F_0 \sim N(0, 0.3^2)$ ,  $G \sim N(0, 10^2)$  and  $\epsilon = 0.3$ . This simple example clearly shows that the estimates based on the  $L_2$  norm with classical CV (bold line) are influenced in a certain region (similar as before) or even breakdown (in case of the gross error model) in contrast to estimates based on robust KBR with robust CV (thin line). The fully robust LS-SVM method will be discussed later in this Section.



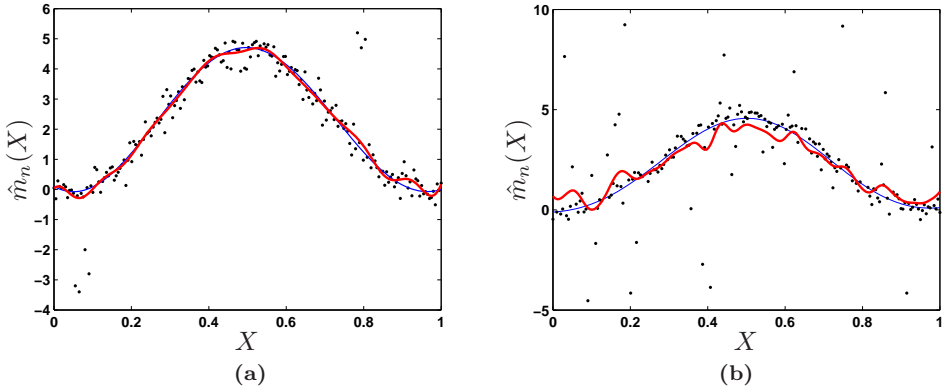
**Figure 5.6:** LS-SVM estimates with (a) normal distributed errors and two groups of outliers; (b) the  $\epsilon$ -contamination model. This clearly shows that the estimates based on the  $L_2$  norm (bold line) are influenced in a certain region or even breakdown in contrast to estimates based on robust loss functions (thin line).

Another important issue to obtain robustness in nonparametric regression is the kernel function  $K$ . Kernels that satisfy  $K(u) \rightarrow 0$  as  $u \rightarrow \infty$ , for  $X \rightarrow \infty$  and  $X \rightarrow -\infty$ , are bounded in  $\mathbb{R}$ . These type of kernels are called decreasing kernels. Using decreasing kernels leads to quite robust methods with respect to outliers in the  $X$ -direction (leverage points). Common choices of decreasing kernels are:  $K(u) = \max((1 - u^2), 0)$ ,  $K(u) = \exp(-u^2)$ ,  $K(u) = \exp(-|u|), \dots$

The last issue to acquire a fully robust estimate is the proper type of cross-validation (CV). When no outliers are present in the data, CV has been shown to produce tuning parameters that are asymptotically consistent (Härdle et al., 1988). Yang (2007) showed that, under some regularity conditions, for an appropriate choice of data splitting ratio, cross-validation is consistent in the sense of selecting the better procedure with probability approaching 1. However, when outliers are present in the data, the use of CV can lead to extremely biased tuning parameters (Leung, 2005) resulting in bad regression estimates. The estimate can also fail when the tuning parameters are determined by standard CV using a robust



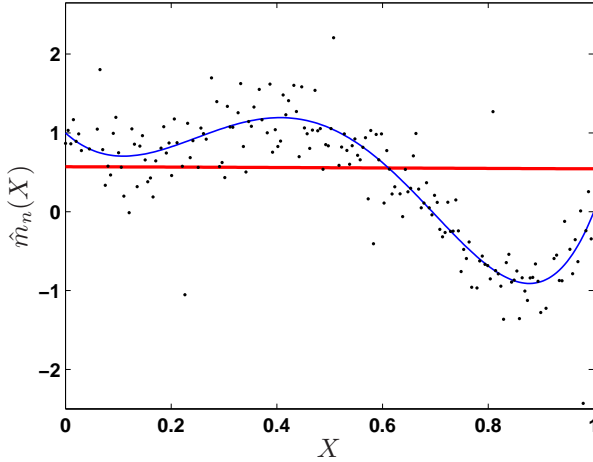
smoother. The reason is that CV no longer produces a reasonable estimate of the prediction error. Therefore, a fully robust CV method is necessary. Figure 5.7 demonstrates this behavior on the same toy example (see Figure 5.6). Indeed, it can be clearly seen that CV results in less optimal tuning parameters resulting in a bad estimate. Hence, to obtain a fully robust estimate, every step has to be robust i.e. robust CV with a robust smoother based on a decreasing kernel.



**Figure 5.7:** LS-SVM estimates and type of errors as in Figure 5.6. The bold line represents the estimate based on classical  $L_2$  CV and a robust smoother. The thin line represents estimates based on a fully robust procedure.

An extreme example to show the absolute necessity of a robust model selection procedure is given next. Consider 200 observations on the interval  $[0,1]$  and a low-order polynomial mean function  $m(X) = 1 - 6X + 36X^2 - 53X^3 + 22X^5$  and  $X \sim \mathcal{U}[0,1]$ . the errors are generated from the gross error model with the same nominal distribution as above and the contamination distribution is taken to be a cubed standard Cauchy with  $\epsilon = 0.3$ . We compare SVM, which is known to be robust, with  $L_2$  CV and the fully robust LS-SVM (robust smoother and robust CV). The result is shown in Figure 5.8. This extreme example confirms the fact that, even if the smoother is robust, also the model selection procedure has to be robust in order to obtain fully robust estimates.

We have demonstrated that fully robust estimates can only be acquired if (i) the smoother is robust, (ii) decreasing kernels are used and (iii) a robust model selection criterion is applied. In what follows we provide some theoretical background on the matter and show how the LS-SVM can be made robust.



**Figure 5.8:** SVM (bold straight line) cannot handle these extreme type of outliers and the estimate becomes useless. The fully robust LS-SVM (thin line) can clearly handle these outliers and does not break down. For visual purposes, not all data is displayed in the figure.

## 5.4.2 Theoretical Background

KBR methods estimate a functional relationship between a dependent variable  $X$  and an independent variable  $Y$ , using a sample of  $n$  observations  $(X_i, Y_i) \in \mathcal{X} \times \mathcal{Y} \subseteq \mathbb{R}^d \times \mathbb{R}$  with joint distribution  $F_{XY}$ . First, we give the following definitions taken from Steinwart and Christmann (2008).

**Definition 5.7 (Steinwart and Christmann, 2008)** *Let  $\mathcal{X}$  be a non-empty set. Then a function  $K : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$  is called a kernel on  $\mathcal{X}$  if there exists a Hilbert space  $\mathcal{H}$  with an inner product  $\langle \cdot, \cdot \rangle$  and a map  $\varphi : \mathcal{X} \rightarrow \mathcal{H}$  such that for all  $x, y \in \mathcal{X}$  we have*

$$K(x, y) = \langle \varphi(x), \varphi(y) \rangle.$$

$\varphi$  is called the feature map and  $\mathcal{H}$  is a feature space of  $K$ .

An example of a frequently used (isotropic) kernel, when  $\mathcal{X} = \mathbb{R}^d$ , is the Gaussian kernel  $K(u) = (1/\sqrt{2\pi}) \exp(-u^2)$  with  $u = \|x - y\|/h$ . Since the Gaussian kernel is an isotropic kernel the notation  $K(x, y) = (1/\sqrt{2\pi}) \exp(-\|x - y\|^2/h^2)$  is the same as  $K(u) = (1/\sqrt{2\pi}) \exp(-u^2)$  with  $u = \|x - y\|/h$ . In this case the feature space  $\mathcal{H}$  is infinite dimensional. Also note that the Gaussian kernel is bounded since

$$\sup_{x, y \in \mathbb{R}^d} K(x, y) = 1.$$

**Definition 5.8 (Steinwart and Christmann, 2008)** Let  $\mathcal{X}$  be a non-empty set and  $\mathcal{H}$  be a Hilbert function space over  $\mathcal{X}$ , i.e. a Hilbert space that consists of functions mapping from  $\mathcal{X}$  into  $\mathbb{R}$ .

- A function  $K : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$  is called a reproducing kernel of  $\mathcal{H}$  if we have  $K(\cdot, x) \in \mathcal{H}$  for all  $x \in \mathcal{X}$  and the reproducing property  $m(x) = \langle m, K(\cdot, x) \rangle$  holds for all  $m \in \mathcal{H}$  and all  $x \in \mathcal{X}$ .
- The space  $\mathcal{H}$  is called a Reproducing Kernel Hilbert Space (RKHS) over  $\mathcal{X}$  if for all  $x \in \mathcal{X}$  the Dirac functional  $\delta_x : \mathcal{H} \rightarrow \mathbb{R}$  defined by

$$\delta_x(m) = m(x), \quad m \in \mathcal{H}$$

is continuous.

Let  $\mathcal{L} : \mathcal{Y} \times \mathbb{R} \rightarrow [0, \infty)$  be a convex loss function. Then the theoretical regularized risk is defined as

$$m_\gamma = \arg \min_{m \in \mathcal{H}} \mathbf{E} [\mathcal{L}(Y, m(X))] + \gamma \|m\|_{\mathcal{H}}^2. \quad (5.5)$$

Before stating the influence function of (5.5) two technical definitions are needed. First, the growth of the loss function  $\mathcal{L}$  is described (Christmann and Steinwart, 2007).

**Definition 5.9 (Christmann and Steinwart, 2007)** Let  $\mathcal{L} : \mathcal{Y} \times \mathbb{R} \rightarrow [0, \infty)$  be a loss function,  $a : \mathcal{Y} \rightarrow [0, \infty)$  be a measurable function and  $p \in [0, \infty)$ . Then  $\mathcal{L}$  is a loss function of type  $(a, p)$  if there exists a constant  $c > 0$  such that

$$\mathcal{L}(y, t) \leq c(a(y) + |t|^p + 1)$$

for all  $y \in \mathcal{Y}$  and all  $t \in \mathbb{R}$ . Furthermore,  $\mathcal{L}$  is of strong type  $(a, p)$  if the first two partial derivatives  $\mathcal{L}'(y, r) = \frac{\partial}{\partial r} \mathcal{L}(y, r)$  and  $\mathcal{L}''(y, r) = \frac{\partial^2}{\partial r^2} \mathcal{L}(y, r)$  of  $\mathcal{L}$  exist and  $\mathcal{L}$ ,  $\mathcal{L}'$  and  $\mathcal{L}''$  are of  $(a, p)$ -type.

Second, we need the following definition about the joint distribution  $F_{XY}$ . For notational ease, we will suppress the subscript  $XY$ .

**Definition 5.10 (Christmann and Steinwart, 2007)** Let  $F$  be a distribution on  $\mathcal{X} \times \mathcal{Y}$ , let  $a : \mathcal{Y} \rightarrow [0, \infty)$  be a measurable function and let  $|F|_a$  be defined as

$$|F|_a = \int_{\mathcal{X} \times \mathcal{Y}} a(y) dF(x, y).$$

If  $a(y) = |y|^p$  for  $p > 0$  we write  $|F|_p$ .

Regarding the theoretical regularized risk (5.5), DeVito et al. (2004) proved the following result.

**Proposition 5.1** *Let  $p = 1$ ,  $\mathcal{L}$  be a convex loss function of strong type  $(a,p)$ , and  $F$  be a distribution on  $\mathcal{X} \times \mathcal{Y}$  with  $|F|_a < \infty$ . Let  $\mathcal{H}$  be the RKHS of a bounded, continuous kernel  $K$  over  $\mathcal{X}$  and  $\varphi : \mathcal{X} \rightarrow \mathcal{H}$  be the feature map of  $\mathcal{H}$ . Then with  $h(x, y) = \mathcal{L}'(y, m_\gamma(x))$  it holds that*

$$m_\gamma = -\frac{1}{2\gamma} \mathbf{E}[h\varphi].$$

Consider the map  $T$  which assigns to every distribution  $F$  on  $\mathcal{X} \times \mathcal{Y}$  with  $|F|_a < \infty$ , the function  $T(F) = m_\gamma \in \mathcal{H}$ . An expression for the influence function (5.2) of  $T$  was proven in Christmann and Steinwart (2007).

**Proposition 5.2** *Let  $\mathcal{H}$  be a RKHS of a bounded continuous kernel  $K$  on  $\mathcal{X}$  with feature map  $\varphi : \mathcal{X} \rightarrow \mathcal{H}$ , and  $\mathcal{L} : \mathcal{Y} \times \mathbb{R} \rightarrow [0, \infty)$  be a convex loss function of some strong type  $(a,p)$ . Furthermore, let  $F$  be a distribution on  $\mathcal{X} \times \mathcal{Y}$  with  $|F|_a < \infty$ . Then the IF of  $T$  exists for all  $z = (z_x, z_y) \in \mathcal{X} \times \mathcal{Y}$  and is given by*

$$\text{IF}(z; T, F) = S^{-1} \{ \mathbf{E}[\mathcal{L}'(Y, m_\gamma(X)) \varphi(X)] \} - \mathcal{L}'(z_y, m_\gamma(z_x)) S^{-1} \varphi(z_x),$$

with  $S : \mathcal{H} \rightarrow \mathcal{H}$  defined as  $S(m) = 2\gamma m + \mathbf{E}[\mathcal{L}''(Y, m_\gamma(X)) \langle \varphi(X), m \rangle \varphi(X)]$ .

From this proposition, it follows immediately that the IF only depends on  $z$  through the term

$$-\mathcal{L}'(z_y, m_\gamma(z_x)) S^{-1} \varphi(z_x).$$

From a robustness point of view, it is important to bound the IF. It is obvious that this can be achieved by using a bounded kernel, e.g. the Gaussian kernel and a loss function with bounded first derivative e.g.  $L_1$  loss or Vapnik's  $\varepsilon$ -insensitive loss. The  $L_2$  loss on the other hand leads to an unbounded IF and hence is not robust.

Although loss functions with bounded first derivative are easy to construct, they lead to more complicated optimization procedures such as QP problems. In case of LS-SVMs this would mean that the  $L_2$  loss should be replaced by e.g. an  $L_1$  loss, what immediately would lead to a QP problem. In what follows we will study an alternative way of achieving robustness by means of reweighting. This has the advantage of easily computable estimates i.e. solving a weighted least squares problem in every iteration. First, we need the following definition concerning the weight function.

**Definition 5.11** *For  $m \in \mathcal{H}$ , let  $V : \mathbb{R} \rightarrow [0, 1]$  be a weight function depending on the residual  $Y - m(X)$  w.r.t.  $m$ . Then the following assumptions will be made on  $V$*

- $V$  is a non-negative bounded Borel measurable function;
- $V$  is an even function of  $r$ ;
- $V$  is continuous and differentiable with  $V'(r) \leq 0$  for  $r > 0$ .

A sequence of successive minimizers of a weighted least squares regularized risk is defined as follows.

**Definition 5.12 (Debruyne et al., 2010)** Let  $m_\gamma^{(0)} \in \mathcal{H}$  be an initial fit, e.g. obtained by ordinary unweighted LS-KBR. Let  $V$  be a weight function satisfying the conditions in Definition 5.11. Then the  $(k+1)^{\text{th}}$  reweighted LS-KBR estimator is defined by

$$m_\gamma^{(k+1)} = \arg \min_{m \in \mathcal{H}} \mathbf{E} \left[ V(Y - m_\gamma^{(k)}(X))(Y - m(X))^2 \right] + \gamma \|m\|_{\mathcal{H}}^2. \quad (5.6)$$

Debruyne et al. (2010) proved that, under certain condition, the IF of reweighted LS-KBR estimator (5.6) is bounded when  $k \rightarrow \infty$  and is given as follows.

**Proposition 5.3** Denote by  $T_{k+1}$  the map  $T_{k+1}(F) = m_\gamma^{(k+1)}$ . Furthermore, let  $F$  be a distribution on  $\mathcal{X} \times \mathcal{Y}$  with  $|F|_2 < \infty$  and  $\int_{\mathcal{X} \times \mathcal{Y}} V(y - m_\gamma^{(\infty)}(x)) dF(x, y) > 0$ . Denote by  $T_\infty$  the map  $T_\infty(F) = m_\gamma^{(\infty)}$ . Denote the operators  $S_{V, \infty} : \mathcal{H} \rightarrow \mathcal{H}$  and  $C_{V, \infty} : \mathcal{H} \rightarrow \mathcal{H}$  given by

$$S_{V, \infty}(m) = \gamma m + \mathbf{E} \left[ V \left( Y - m_\gamma^{(\infty)}(X) \right) \langle m, \varphi(X) \rangle \varphi(X) \right]$$

and

$$C_{V, \infty}(m) = - \mathbf{E} \left[ V' \left( Y - m_\gamma^{(\infty)}(X) \right) \left( Y - m_\gamma^{(\infty)}(X) \right) \langle m, \varphi(X) \rangle \varphi(X) \right].$$

Further, assume that  $\|S_{V, \infty}^{-1} \circ C_{V, \infty}\| < 1$ . Then the IF of  $T_\infty$  exists for all  $z = (z_x, z_y) \in \mathcal{X} \times \mathcal{Y}$  and is given by

$$\begin{aligned} \text{IF}(z; T_\infty, F) &= (S_{V, \infty} - C_{V, \infty})^{-1} \left\{ - \mathbf{E} \left[ V \left( Y - m_\gamma^{(\infty)}(X) \right) \left( Y - m_\gamma^{(\infty)}(X) \right) \varphi(X) \right] \right. \\ &\quad \left. + V \left( z_y - m_\gamma^{(\infty)}(z_x) \right) \left( z_y - m_\gamma^{(\infty)}(z_x) \right) \varphi(z_x) \right\}. \end{aligned}$$

The condition  $\|S_{V, \infty}^{-1} \circ C_{V, \infty}\| < 1$  is needed to ensure that the IF of the initial estimator eventually disappears. Notice that the operators  $S_{V, \infty}$  and  $C_{V, \infty}$  are independent of the contamination  $z$ . Since  $\|\varphi(x)\|_{\mathcal{H}}^2 = \langle \varphi(x), \varphi(x) \rangle = K(x, x)$ , the  $\text{IF}(z; T_\infty, F)$  is bounded if

$$\|V(r)r\varphi(x)\|_{\mathcal{H}} = w(r)|r|\sqrt{K(x, x)}$$

is bounded for all  $(x, r) \in \mathbb{R}^d \times \mathbb{R}$ . From Proposition 5.3, the following result immediately follows

**Corollary 5.1** *Assume that the conditions of Proposition 5.3 and Definition 5.11 are satisfied, then  $\|\text{IF}(z; T_\infty, F)\|_{\mathcal{H}}$  bounded implies  $\|\text{IF}(z; T_\infty, F)\|_\infty$  bounded for bounded kernels.*

PROOF. For any  $m \in \mathcal{H} : \|m\|_\infty \leq \|m\|_{\mathcal{H}} \|K\|_\infty$ . The result immediately follows for a bounded kernel  $K$ .  $\square$

An interesting fact which has practical consequences is the choice of the kernel function. It is readily seen that if one takes a Gaussian kernel, only downweighting the residual is needed as the influence in the  $X$ -space is controlled by the kernel. On the other hand, taking an unbounded kernel such as the linear or polynomial kernel requires a weight function that decreases with the residual as well as with  $x$  to obtain a bounded IF. See also Dollinger and Staudte (1991) for similar results regarding ordinary LS and Jorgensen (1993) for iteratively defined statistics.

It does not suffice to derive the IF of the reweighted LS-KBR but also to establish conditions for convergence. The following proposition is due to Debruyne et al. (2010).

**Proposition 5.4 (Conditions for Convergence)** *Define  $V(r) = \frac{\psi(r)}{r}$  with  $\psi$  the contrast function. Then, reweighted LS-KBR with a bounded kernel converges to a bounded influence, even if the initial LS-KBR is not robust, if*

- (c1)  $\psi : \mathbb{R} \rightarrow \mathbb{R}$  is a measurable, real, odd function;
- (c2)  $\psi$  is continuous and differentiable;
- (c3)  $\psi$  is bounded;
- (c4)  $\mathbf{E}_{F_e} \psi'(e) > -\gamma$  where  $F_e$  denotes the distribution of the errors.

Finally, Debruyne et al. (2010) pointed out that reweighting is not only useful when outliers are present in the data but it also leads to a more stable method, especially at heavy tailed distributions. Debruyne et al. (2010) introduced the following stability criterion based on the IF

$$\sup_{i \in \{1, \dots, n\}} \frac{|\text{IF}(z_i; T, F)|}{n} \rightarrow 0. \quad (5.7)$$

If a method is robust, then its IF is bounded over all possible points  $z$  in the support of  $F$  and hence (5.7) is obviously satisfied. The speed of convergence of (5.7) is of order  $O(\log n/n)$  for Gaussian distributed noise. For more heavy tailed distributions, this rate will be much slower.

### 5.4.3 Application to Least Squares Support Vector Machines

A first attempt to robustify LS-SVM was introduced by Suykens et al. (2002). Their approach is based on weighting the residuals from the unweighted LS-SVM (one time). However, from the above theoretical results we know that reweighting only once does not guarantee that the IF is bounded. In order to bound the IF, we have to repeat the weighting procedure a number of times.

Given a data set  $\mathcal{D}_n = \{(X_1, Y_1), \dots, (X_n, Y_n)\}$ . The weighted LS-SVM is formulated as follows

$$\begin{aligned} \min_{w,b,e} \mathcal{J}(w,e) &= \frac{1}{2}w^T w + \frac{\gamma}{2} \sum_{k=1}^n v_k e_k^2 \\ \text{s.t. } Y_k &= w^T \varphi(X_k) + b + e_k, \quad k = 1, \dots, n, \end{aligned} \quad (5.8)$$

where  $v_k$  denotes the weight of the  $k^{\text{th}}$  residual. Again, by using Lagrange multipliers, the solution to (5.8) in the dual variables  $\alpha$  is given by solving the linear system

$$\left( \begin{array}{c|c} 0 & 1_n^T \\ \hline 1_n & \Omega + D_\gamma \end{array} \right) \begin{pmatrix} b \\ \alpha \end{pmatrix} = \begin{pmatrix} 0 \\ Y \end{pmatrix}, \quad (5.9)$$

with  $D_\gamma = \text{diag} \left\{ \frac{1}{\gamma v_1}, \dots, \frac{1}{\gamma v_n} \right\}$ ,  $Y = (Y_1, \dots, Y_n)^T$ ,  $1_n = (1, \dots, 1)^T$ ,  $\alpha = (\alpha_1, \dots, \alpha_n)^T$  and  $\Omega_{kl} = \varphi(X_k)^T \varphi(X_l) = K(X_k, X_l)$  for  $k, l = 1, \dots, n$  and  $K$  a positive definite bounded kernel.

Based on the previous LS-SVM solutions, using an iteratively reweighting approach, a robust estimate can be obtained. In the  $i^{\text{th}}$  iteration, one weighs the error variables  $\hat{e}_k^{(i)} = \hat{\alpha}_k^{(i)} / \gamma$  for  $k = 1, \dots, n$  with weighting factors  $v^{(i)} = (v_1^{(i)}, \dots, v_n^{(i)})^T \in \mathbb{R}^n$ , determined by a weight function  $V$ . Hence, one obtains an iterative algorithm (Algorithm 5) to obtain a robust estimate. This type of

---

#### Algorithm 5 Iteratively Reweighted LS-SVM

---

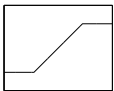
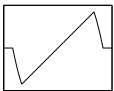
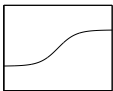
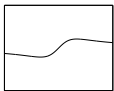
- 1: Compute the residuals  $\hat{e}_k = \hat{\alpha}_k / \gamma$  from the unweighted LS-SVM ( $v_k = 1, \forall k$ )
  - 2: **repeat**
  - 3:   Compute  $\hat{s} = 1.483 \text{MAD}(e_k^{(i)})$  from the  $e_k^{(i)}$  distribution
  - 4:   Choose a weight function  $V$  and determine weights  $v_k^{(i)}$  based on  $r^{(i)} = e_k^{(i)} / \hat{s}$ ;
  - 5:   Solve (5.9) with  $D_\gamma = \text{diag} \left\{ 1/(\gamma v_1^{(i)}), \dots, 1/(\gamma v_n^{(i)}) \right\}$ ,
  - 6:   Set  $i = i + 1$
  - 7: **until** consecutive estimates  $\alpha_k^{(i-1)}$  and  $\alpha_k^{(i)}$  are sufficiently close to each other,  
e.g.  $\max_k (|\alpha_k^{(i-1)} - \alpha_k^{(i)}|) \leq 10^{-4}$ .
-

algorithm can be applied to any kernel based smoother based on an  $L_2$  loss, e.g. local polynomial regression, NW, . . . , to result in a robust estimate. Naturally, it can also be extended to FS-LSSVM (see Chapter 4). The selection of the prototype vectors will not change when outliers are present in the data set since it considers only the values in the  $X$ -space. Even if leverage points are present, the use of bounded kernels in the Nyström approximation will control the influence in the  $X$ -space. Therefore, no noteworthy adaptations are needed in order to robustify FS-LSSVM.

### 5.4.4 Weight Functions

It is without doubt that the choice of weight function  $V$  plays a significant role in the robustness aspects of the smoother. We will show later that the choice of weight function also has an influence on the speed of convergence. We consider four different weight functions illustrated in Table 5.1.

**Table 5.1:** Definitions for the Huber, Hampel, Logistic and Myriad weight functions  $V(\cdot)$ . The corresponding loss  $\mathcal{L}(\cdot)$  and score function  $\psi(\cdot)$  are also given.

	Huber	Hampel	Logistic	Myriad
$V(r)$	$\begin{cases} 1, & \text{if }  r  < \beta; \\ \frac{\beta}{ r }, & \text{if }  r  \geq \beta. \end{cases}$	$\begin{cases} 1, & \text{if }  r  < b_1; \\ \frac{b_2 -  r }{b_2 - b_1}, & \text{if } b_1 \leq  r  \leq b_2; \\ 0, & \text{if }  r  > b_2. \end{cases}$	$\frac{\tanh(r)}{r}$	$\frac{\delta^2}{\delta^2 + r^2}$
$\psi(r)$				
$\mathcal{L}(r)$	$\begin{cases} r^2, & \text{if }  r  < \beta; \\ \beta r  - \frac{\beta^2}{2}, & \text{if }  r  \geq \beta. \end{cases}$	$\begin{cases} r^2, & \text{if }  r  < b_1; \\ \frac{b_2 r^2 -  r ^3}{b_2 - b_1}, & \text{if } b_1 \leq  r  \leq b_2; \\ 0, & \text{if }  r  > b_2. \end{cases}$	$r \tanh(r)$	$\log(\delta^2 + r^2)$

The first three are well-known in the field of robust statistics, while the last one is less or not known. We introduce some of the properties of the last weight function i.e. the Myriad (see Arce (2005) for applications of Myriad filters in signal processing). The Myriad is derived from the Maximum Likelihood (ML) estimation of a Cauchy distribution with scaling factor  $\delta$  (see below) and can be used as a robust location estimator in stable noise environments. Given a set of i.i.d. random variables  $X_1, \dots, X_n \sim X$  and  $X \sim C(\zeta, \delta)$ , where the location parameter  $\zeta$  is to be estimated from data i.e.  $\hat{\zeta}$  and  $\delta > 0$  is a scaling factor. The



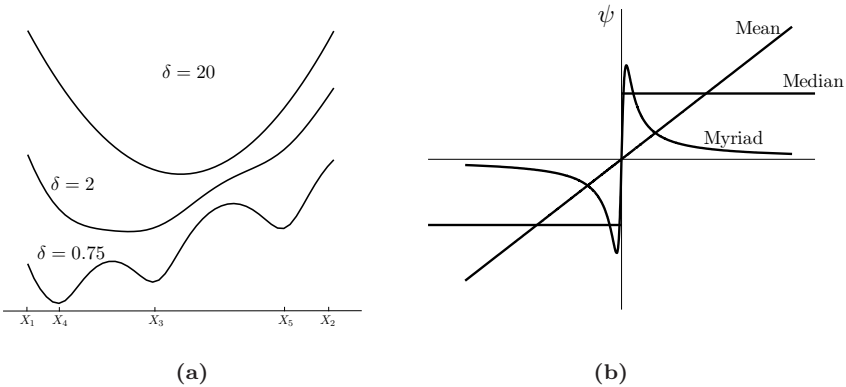
ML principle yields the sample Myriad

$$\hat{\zeta}_\delta = \arg \max_{\zeta \in \mathbb{R}} \left( \frac{\delta}{\pi} \right)^n \prod_{i=1}^n \frac{1}{\delta^2 + (X_i - \zeta)^2},$$

which is equivalent to

$$\hat{\zeta}_\delta = \arg \min_{\zeta \in \mathbb{R}} \sum_{i=1}^n \log [\delta^2 + (X_i - \zeta)^2]. \tag{5.10}$$

Note that, unlike the sample mean or median, the definition of the sample Myriad involves the free parameter  $\delta$ . We will refer to  $\delta$  as the linearity parameter of the Myriad. The behavior of the Myriad estimator is markedly dependent on the value of its linearity parameter  $\delta$ . Tuning the linearity parameter  $\delta$  adapts the behavior of the myriad from impulse-resistant mode-type estimators (small  $\delta$ ) to the Gaussian-efficient sample mean (large  $\delta$ ). If an observation in the set of input samples has a large magnitude such that  $|X_i - \zeta| \gg \delta$ , the cost associated with this sample is approximately  $\log(X_i - \zeta)^2$  i.e. the log of squared deviation. Thus, much as the sample mean and sample median respectively minimize the sum of square and absolute deviations, the sample myriad (approximately) minimizes the sum of logarithmic squared deviations. Some intuition can be gained by plotting the cost function (5.10) for various values of  $\delta$ . Figure 5.9a depicts the different cost function characteristics obtained for  $\delta = 20, 2, 0.75$  for a sample set of size 5. For a set of samples defined as above, an M-estimator of location is defined as the parameter  $\zeta$  minimizing a sum of the form  $\sum_{i=1}^n \mathcal{L}(X_i - \zeta)$ , where  $\mathcal{L}$  is



**Figure 5.9:** (a) Myriad cost functions for the observation samples  $X_1 = -3, X_2 = 8, X_3 = 1, X_4 = -2, X_5 = 5$  for  $\delta = 20, 2, 0.2$ ; (b) Influence function for the mean, median and Myriad.

the cost or loss function. In general, when  $\mathcal{L}(x) = -\log f(x)$ , with  $f$  a density, the M-estimate  $\hat{\zeta}$  corresponds to the ML estimator associated with  $f$ . According to (5.10), the cost function associated with the sample Myriad is given by

$$\mathcal{L}(x) = \log[\delta^2 + x^2].$$

Some insight in the operation of M-estimates is gained through the definition of the IF. For an M-estimate, the IF is proportional to the score function (Hampel et al., 1986, p. 101). For the Myriad (see also Figure 5.9b), the IF is given by

$$\mathcal{L}'(x) = \psi(x) = \frac{2x}{\delta^2 + x^2}.$$

When using the Myriad as a location estimator, it can be shown that the Myriad offers a rich class of operation modes that can be controlled by varying the parameter  $\delta$ . When the noise is Gaussian, large values of  $\delta$  can provide the optimal performance associated with the sample mean, whereas for highly impulsive noise statistics, the resistance of mode-type estimators can be achieved by setting low values of  $\delta$ . Also, the Myriad has a linearity property i.e. when  $\delta \rightarrow \infty$  then the sample Myriad reduces to the sample mean.

**Theorem 5.1 (Linearity Property)** *Given a set of samples  $X_1, \dots, X_n$ . The sample Myriad  $\hat{\zeta}_\delta$  converges to the sample mean as  $\delta \rightarrow \infty$ , i.e.*

$$\hat{\zeta}_\infty = \lim_{\delta \rightarrow \infty} \hat{\zeta}_\delta = \lim_{\delta \rightarrow \infty} \left\{ \arg \min_{\zeta \in \mathbb{R}} \sum_{i=1}^n \log [\delta^2 + (X_i - \zeta)^2] \right\} = \frac{1}{n} \sum_{i=1}^n X_i.$$

PROOF. First, we establish upper and lower bounds for  $\hat{\zeta}_\delta$ . Consider the order statistic  $X_{(1)} \leq \dots \leq X_{(n)}$  of the sample  $X_1, \dots, X_n$ . Then, by taking  $\zeta < X_{(1)} = \min\{X_1, \dots, X_n\}$  and for all  $i$

$$\delta^2 + (X_i - X_{(1)})^2 < \delta^2 + (X_i - \zeta)^2,$$

it follows that  $\hat{\zeta}_\delta \geq X_{(1)}$ . Similarly, one can find that  $\hat{\zeta}_\delta \leq X_{(n)}$ . Hence,

$$\begin{aligned} \hat{\zeta}_\delta &= \arg \min_{X_{(1)} \leq \zeta \leq X_{(n)}} \prod_{i=1}^n [\delta^2 + (X_i - \zeta)^2] \\ &= \arg \min_{X_{(1)} \leq \zeta \leq X_{(n)}} \delta^{2n} + \delta^{2n-2} \sum_{i=1}^n (X_i - \zeta)^2 + O(\delta^{2n-4}) \\ &= \arg \min_{X_{(1)} \leq \zeta \leq X_{(n)}} \sum_{i=1}^n (X_i - \zeta)^2 + \frac{O(\delta^{2n-4})}{\delta^{2n-2}}. \end{aligned}$$

For  $\delta \rightarrow \infty$  the last term becomes negligible and

$$\hat{\zeta}_\infty \rightarrow \arg \min_{X_{(1)} \leq \zeta \leq X_{(n)}} \sum_{i=1}^n (X_i - \zeta)^2 = \frac{1}{n} \sum_{i=1}^n X_i.$$

□

As the Myriad moves away from the linear region (large values of  $\delta$ ) to lower values of  $\delta$ , the estimator becomes more resistant to outliers. When  $\delta$  tends to zero, the myriad approaches the mode of the sample.

**Theorem 5.2 (Mode Property)** *Given a set of samples  $X_1, \dots, X_n$ . The sample Myriad  $\hat{\zeta}_\delta$  converges to a mode estimator for  $\delta \rightarrow 0$ . Further,*

$$\hat{\zeta}_0 = \lim_{\delta \rightarrow 0} \hat{\zeta}_\delta = \arg \min_{X_j \in \mathcal{K}} \prod_{X_i \neq X_j}^n |X_i - X_j|,$$

where  $\mathcal{K}$  is the set of most repeated values.

PROOF. Since  $\delta > 0$ , the sample Myriad (5.10) can be written as

$$\arg \min_{\zeta \in \mathbb{R}} \prod_{i=1}^n \left[ 1 + \frac{(X_i - \zeta)^2}{\delta^2} \right].$$

For small values of  $\delta$ , the first term in the sum, i.e. 1, can be omitted, hence

$$\prod_{i=1}^n \left[ 1 + \frac{(X_i - \zeta)^2}{\delta^2} \right] = O \left( \frac{1}{\delta^2} \right)^{n - \kappa(\zeta)}, \tag{5.11}$$

where  $\kappa(\zeta)$  is the number of times that  $\zeta$  is repeated in the sample  $X_1, \dots, X_n$ . The right-hand side of (5.11) is minimized for  $\zeta$  when the exponent  $n - \kappa(\zeta)$  is minimized. Therefore,  $\hat{\zeta}_0$  will be a maximum of  $\kappa(\zeta)$  and consequently,  $\hat{\zeta}_0$  will be the most repeated value in the sample  $X_1, \dots, X_n$  or the mode.

Let  $\kappa = \max_j \kappa(X_j)$  and  $X_j \in \mathcal{K}$ . Then,

$$\prod_{X_i \neq X_j}^n \left[ 1 + \frac{(X_i - X_j)^2}{\delta^2} \right] = \prod_{X_i \neq X_j}^n \left[ \frac{(X_i - X_j)^2}{\delta^2} \right] + O \left( \frac{1}{\delta^2} \right)^{(n - \kappa) - 1}. \tag{5.12}$$

For small  $\delta$ , the second term in (5.12) will be small compared to the first term, since this is of order  $O\left(\frac{1}{\delta^2}\right)^{n-\kappa}$ . Finally,  $\hat{\zeta}_0$  can be computed as follows.

$$\begin{aligned}\hat{\zeta}_0 &= \arg \min_{X_j \in \mathcal{K}} \prod_{X_i \neq X_j}^n \left[ \frac{(X_i - X_j)^2}{\delta^2} \right] \\ &= \arg \min_{X_j \in \mathcal{K}} \prod_{X_i \neq X_j}^n |X_i - X_j|.\end{aligned}$$

□

### 5.4.5 Speed of Convergence-Robustness Tradeoff

Debruyne et al. (2010) established conditions for convergence in case of reweighted LS-KBR, see Proposition 5.4. Define

$$d = \mathbf{E}_{F_e} \frac{\psi(e)}{e} \quad \text{and} \quad c = d - \mathbf{E}_{F_e} \psi'(e),$$

then  $c/d$  establishes an upper bound on the reduction of the influence function at each step (Debruyne et al., 2010). The upper bound represents a trade-off between the reduction of the influence function (speed of convergence) and the degree of robustness. The higher the ratio  $c/d$ , the higher the degree of robustness but the slower the reduction of the influence function at each step and vice versa. In Table 5.2 this upper bound is calculated for a Normal distribution and a standard Cauchy for the four types of weighting schemes. Note that the convergence of the influence function is quite fast, even at heavy tailed distributions. For Huber and Myriad weights, the convergence rate decreases rapidly as  $\beta$  respectively  $\delta$  increases. This behavior is to be expected, since the larger  $\beta$  respectively  $\delta$ , the less points are downweighted. Also note that the upper bound on the convergence rate approaches 1 as  $\beta, \delta \rightarrow 0$ , indicating a high degree of robustness but slow convergence rate. Therefore, logistic weights offer a good tradeoff between speed of convergence and degree of robustness. Also notice the small ratio for the Hampel weights indicating a low degree of robustness. The highest degree of robustness is achieved by using Myriad weights.

### 5.4.6 Robust Selection of Tuning Parameters

It is shown in Figure 5.7 that also the model selection procedure plays a significant role in obtaining fully robust estimates. Leung (2005) theoretically shows that a robust CV procedure differs from the Mean Asymptotic Squared Error (MASE)

**Table 5.2:** Values of the constants  $c$ ,  $d$  and  $c/d$  for the Huber, Logistic, Hampel and Myriad weight function at a standard Normal distribution and a standard Cauchy. The bold values represent an upper bound for the reduction of the influence function at each step.

Weight function	Parameter settings	$N(0,1)$			$C(0,1)$		
		$c$	$d$	$c/d$	$c$	$d$	$c/d$
Huber	$\beta = 0.5$	0.32	0.71	<b>0.46</b>	0.26	0.55	<b>0.47</b>
	$\beta = 1$	0.22	0.91	<b>0.25</b>	0.22	0.72	<b>0.31</b>
Logistic		0.22	0.82	<b>0.26</b>	0.21	0.66	<b>0.32</b>
Hampel	$b_1 = 2.5$ $b_2 = 3$	0.006	0.99	<b>0.006</b>	0.02	0.78	<b>0.025</b>
Myriad	$\delta = 0.1$	0.11	0.12	<b>0.92</b>	0.083	0.091	<b>0.91</b>
	$\delta = 1$	0.31	0.66	<b>0.47</b>	0.25	0.50	<b>0.50</b>

by a constant shift and a constant multiple. Neither of these are dependent on the bandwidth. Further, it is shown that this multiple depends on the score function and therefore, also on the weight function. To obtain a fully robust procedure for LS-KBR one needs also, besides a robust smoother and bounded kernel, a robust model selection criterion. Consider for example the robust LOO-CV (RLOO-CV) given by

$$\text{RLOO-CV}(\theta) = \frac{1}{n} \sum_{i=1}^n \mathcal{L} \left( Y_i, \hat{m}_{n,\text{rob}}^{(-i)}(X_i; \theta) \right), \tag{5.13}$$

where  $\mathcal{L}$  is a robust loss function e.g.  $L_1$ , Huber loss, Myriad loss,  $\hat{m}_{n,\text{rob}}$  is a robust smoother and  $\hat{m}_{n,\text{rob}}^{(-i)}(X_i; \theta)$  denotes the leave-one-out estimator where point  $i$  is left out from the training and  $\theta$  denotes the tuning parameter vector, e.g. when using Myriad weights  $\theta = (h, \gamma, \delta)$ . A similar principle can be used in robust  $v$ -fold CV. For robust counterparts of GCV and complexity criteria see e.g. Lukas (2008), Ronchetti (1985) and Burman and Nolan (1995). De Brabanter (2004, Chapter 11) transformed the robust CV as a location estimation problem and used  $L$ -estimators (Daniell, 1920) (trimmed mean and Winsorized mean) to achieve robustness.

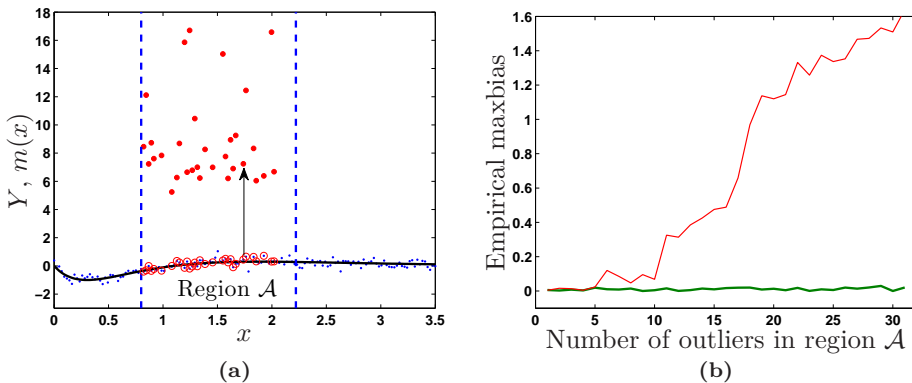
## 5.5 Simulations

### 5.5.1 Empirical Maxbias Curve

We compute the empirical maxbias curve (5.3) for both LS-SVM and its robust counterpart iteratively reweighted (IRLS-SVM) on a test point. Given 150 “good” equispaced observations according to the relation

$$Y_k = m(x_k) + e_k, \quad k = 1, \dots, 150,$$

where  $e_k \sim \mathcal{N}(0, 0.1^2)$  and  $m(x_k) = 4.26 [\exp(-x_k) - 4 \exp(-2x_k) + 3 \exp(-3x_k)]$  (Wahba, 1990, Chapter 4, p. 45). Let  $\mathcal{A} = \{x : 0.8 \leq x \leq 2.22\}$  denote a particular region (consisting of 60 data points) and let  $x = 1.5$  be a test point in that region. In each step, we start to contaminate the region  $\mathcal{A}$  by deleting one “good” observation and replacing it by a “bad” point  $(x_k, Y_k^b)$ , see Figure 5.10a. In each step, the value  $Y_k^b$  is chosen as the absolute value of a standard Cauchy random variable. We repeat this until the estimation becomes useless. A maxbias plot is shown in Figure 5.10b where the values of the non-robust LS-SVM estimate  $\hat{m}_n(x)$  and the robust IRLS-SVM estimate  $\hat{m}_{n,\text{rob}}(x)$  are drawn as a function of the number of outliers in region  $\mathcal{A}$ . The tuning parameters are tuned with  $L_2$  LOO-CV for LS-SVM and RLOO-CV (5.13), based on an  $L_1$  loss and Myriad weights, for IRLS-SVM. The maxbias curve of  $\hat{m}_{n,\text{rob}}(x)$  increases very slightly with the number of outliers in region  $\mathcal{A}$  and stays bounded right up to the breakdown point. This is in strong contrast with the non-robust LS-SVM estimate  $\hat{m}_n(x)$  which has a breakdown point equal to zero.

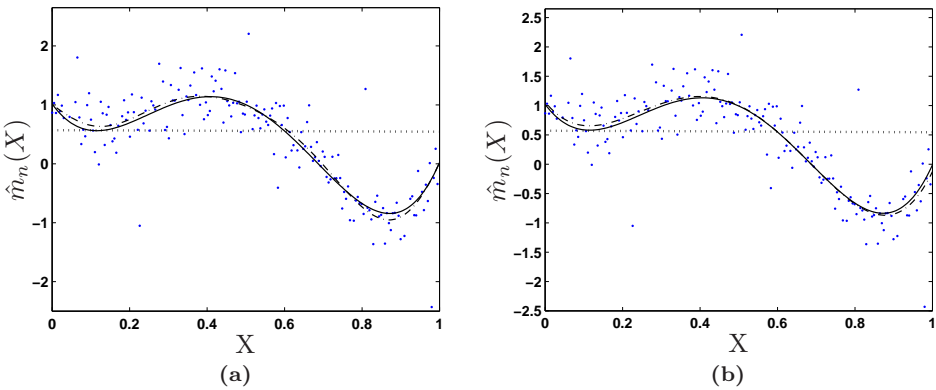


**Figure 5.10:** (a) In each step, one good point (circled dots) of the the region  $\mathcal{A} = \{x : 0.8 \leq x \leq 2.22\}$  is contaminated by the absolute value of a standard Cauchy random variable (full dots) until the estimation becomes useless; (b) Empirical maxbias curve of the non-robust LS-SVM estimator  $\hat{m}_n(x)$  (thine line) and IRLS-SVM estimator  $\hat{m}_{n,\text{rob}}(x)$  (bold line) in a test point  $x = 1.5$ .

### 5.5.2 Toy example

Recall the low order polynomial function, in the beginning of this Section, with 200 observations according to  $m(X) = 1 - 6X + 36X^2 - 53X^3 + 22X^5$  and  $X \sim U[0,1]$ . The distribution of the errors is given by the gross error model with  $\epsilon = 0.3$ ,  $F_0 = N(0,0.1)$  and  $G = C^3(0,1)$ . The results for the four types of weight functions (see Table 5.1) are shown in Figure 5.11 and performances in the three norms are given in Table 5.3. For this simulation we set  $\beta = 1.345$ ,  $b_1 = 2.5$  and  $b_2 = 3$  and  $\delta$  is tuned via 10-fold robust CV (5.13) with  $L_1$  loss. For all simulations, the learning parameters are tuned via 10-fold robust cross-validation with  $L_1$  loss. This simulation shows that the four weight functions are able to handle these extreme outliers. It is clear that the Myriad weight function outperforms the others. This is to be expected since it was designed for such types of outliers.

Due to the non-robust CV procedure, used to find the tuning parameters of SVM, the estimate breaks down even when SVMs are robust. The simulation confirms the theoretical justifications, saying that three requirements have to be satisfied in order to achieve a fully robust method.



**Figure 5.11:** Low order polynomial function with 200 observations according to  $f(X) = 1 - 6X + 36X^2 - 53X^3 + 22X^5$  and  $X \sim U[0,1]$ . The distribution of the errors is given by the gross error model with  $\epsilon = 0.3$ ,  $F_0 = N(0,0.1)$  and  $G = C^3(0,1)$ . The dotted line is the corresponding SVM fit (tuned with  $L_2$  CV). The iteratively reweighted LS-SVM with (a) Huber weights (full line) and Hampel weights (dash dotted line); (b) Logistic weights (full line) and Myriad weights (dash dotted line). For visual purposes, not all data is displayed in the figure.

**Table 5.3:** Performances in the three norms (difference between the estimated function and the true underlying function) of the different weight functions used in iteratively reweighted LS-SVM on the low order polynomial. The last column denotes the number of iterations  $i_{\max}$  needed to satisfy the stopping criterion in Algorithm 5.

	$L_1$	$L_2$	$L_\infty$	$i_{\max}$
Huber	0.06	0.005	0.12	7
Hampel	0.06	0.005	0.13	4
Logistic	0.06	0.005	0.11	11
Myriad	0.03	0.002	0.06	17

### 5.5.3 Real Life Data Sets

The octane data (Hubert et al., 2005) consist of NIR absorbance spectra over 226 wavelengths ranging from 1102 to 1552 nm. For each of the 39 production gasoline samples the octane number was measured. It is well known that the octane data set contains six outliers to which alcohol was added. Table 5.4 shows the result (median and median absolute deviation for each method are reported) of a Monte Carlo simulation (200 runs) of the iteratively reweighted LS-SVM (IRLS-SVM), weighted LS-SVM (WLS-SVM) (Suykens et al., 2002) (based on Hampel weights) and SVM in different norms on a randomly chosen test set of size 10. Model selection was performed using robust LOO-CV.

As a next example consider the data about the demographical information on the 50 states of the USA in 1980. The data set provides information on 25 variables. The goal is to determine the murder rate per 100,000 population. The result is shown in Table 5.4 for randomly chosen test sets of size 15. The results of the simulations show that by using reweighting schemes the performance can be improved over weighted LS-SVM and SVM. To illustrate the trade-off between the degree of robustness and speed of convergence, the number of iterations  $i_{\max}$  are also given in Table 5.4. The stopping criterion was taken identically to the one in Algorithm 5. The number of iterations, needed by each weight function, confirms the results in Table 5.2.



**Table 5.4:** Results on the Octane and Demographic data sets. For 200 simulations the medians and median absolute deviations (between brackets) of three norms are given (on test data).  $i_{\max}$  denotes the number of iterations needed to satisfy the stopping criterion in Algorithm 5. The best results are bold faced.

		Octane				Demographic			
	weights	$L_1$	$L_2$	$L_\infty$	$i_{\max}$	$L_1$	$L_2$	$L_\infty$	$i_{\max}$
IRLS	Huber	<b>0.19</b> (0.03)	0.07 (0.02)	0.51 (0.10)	15	0.31 (0.01)	0.14 (0.02)	0.83 (0.06)	8
	Hampel	0.22 (0.03)	0.07 (0.03)	0.55 (0.14)	2	0.33 (0.01)	0.18 (0.04)	0.97 (0.02)	3
SVM	Logistic	0.20 (0.03)	<b>0.06</b> (0.02)	0.51 (0.10)	18	0.30 (0.02)	<b>0.13</b> (0.01)	0.80 (0.07)	10
	Myriad	0.20 (0.03)	<b>0.06</b> (0.02)	<b>0.50</b> (0.09)	22	<b>0.30</b> (0.01)	<b>0.13</b> (0.01)	<b>0.79</b> (0.06)	12
WLS SVM		0.22 (0.03)	0.08 (0.02)	0.60 (0.15)	1	0.33 (0.02)	0.15 (0.01)	0.80 (0.02)	1
SVM		0.28 (0.03)	0.12 (0.02)	0.56 (0.13)	-	0.37 (0.02)	0.21 (0.02)	0.90 (0.06)	-

## 5.6 Conclusions

In this Chapter, we reviewed some measures of robustness and applied these measures to simple statistics such as the mean, median and variance. We discussed the different approaches used in the literature for achieving robustness in parametric and nonparametric regression models. Further, we illustrated how robustness in the nonparametric case can be obtained by using a least squares cost function. Also, we showed, in order to achieve a fully robust procedure, three requirements have to be fulfilled i.e. (i) robust smoother, (ii) bounded kernel and (iii) a robust model selection procedure. Finally, we obtained a robust LS-SVM estimator via iterative reweighting. We compared four different weight functions and investigated their application in iteratively reweighted LS-SVM. We introduced the Myriad reweighting and derived its linear and mode property. We demonstrated that, by means of simulations and theoretical results, reweighting is useful not only when outliers are present in the data but also to improve stability, especially at heavy tailed distributions. By means of an upper bound for the reduction of the influence function in each step, we revealed the existence of a tradeoff between speed of convergence and the degree of robustness. We

demonstrated that the Myriad weight function is highly robust against (extreme) outliers but exhibits a slow speed of convergence. A good compromise between the speed of convergence and robustness can be achieved by using Logistic weights.

# Chapter 6

## Kernel Regression with Correlated Errors

In all previous Chapters, i.i.d. data is considered. In this Chapter, we will investigate the consequences when this assumption is violated. We will show that, for nonparametric kernel based regression, the model selection procedures such as LOO-CV, GCV,  $v$ -fold CV break down in the presence of correlated data rather than the smoothing method. In order to cope with correlated data, we prove that a kernel  $K$  satisfying  $K(0) = 0$  removes the correlation structure without requiring any prior knowledge about its structure. Finally, we will show that the form of the kernel, based on mean squared error, is very important when errors are correlated. Contributions are made in Section 6.3.

### 6.1 Introduction

From the previous Chapters, we can conclude that nonparametric regression is a very popular tool for data analysis because these techniques impose few assumptions about the shape of the mean function. Hence, they are extremely flexible tools for uncovering nonlinear relationships between variables. Given the data  $\{(x_1, Y_1), \dots, (x_n, Y_n)\}$  where  $x_i \equiv i/n$  and  $x \in [0, 1]$ . Then, the data can be written as

$$Y_i = m(x_i) + e_i, \quad i = 1, \dots, n, \quad (6.1)$$

where  $e_i = Y_i - m(x_i)$  satisfies  $\mathbf{E}(e) = 0$  and  $\mathbf{Var}(e) = \sigma^2 < \infty$ . Thus  $Y_i$  can be considered as the sum of the value of the regression function at  $x_i$  and some error  $e_i$  with the expected value zero and the sequence  $\{e_i\}$  is a covariance stationary process.

**Definition 6.1 (Covariance Stationarity)** *The sequence  $\{e_i\}$  is covariance stationary if*

- $\mathbf{E}[e_i] = \mu$  for all  $i$ ;
- $\mathbf{Cov}[e_i, e_{i-j}] = \mathbf{E}[(e_i - \mu)(e_{i-j} - \mu)] = \gamma_j$  for all  $i$  and any  $j$ .

Many techniques include a smoothing parameter and/or kernel bandwidth which controls the smoothness, bias and variance of the estimate. A vast number of techniques (see Chapter 3) have been developed to determine suitable choices for these tuning parameters from data when the errors are independent and identically distributed (i.i.d.) with finite variance. More detailed information can be found in the books of Fan and Gijbels (1996), Davison and Hinkley (2003) and Konishi and Kitagawa (2008) and the article by Feng and Heiler (2009). However, all the previous techniques have been derived under the i.i.d. assumption. It has been shown that violating this assumption results in the break down of the above methods (Altman, 1990; Hermann et al., 1992; Opsomer et al., 2001; Lahiri, 2003). If the errors are positively (negatively) correlated, these methods will produce a small (large) bandwidth which results in a rough (oversmooth) estimate of the regression function. The focus of this Chapter is to look at the problem of estimating the mean function  $m$  in the presence of correlation, not that of estimating the correlation function itself. Approaches describing the estimation of the correlation function are extensively studied in Hart and Wehrly (1986), Hart (1991) and Park et al. (2006).

Another issue in this context is whether the errors are assumed to be short-range dependent, where the correlation decreases rapidly as the distance between two observations increases or long-range dependent. The error process is said to be short-range dependent if for some  $\tau > 0$ ,  $\delta > 1$  and correlation function  $\rho(\cdot)$ , the spectral density  $H(\omega) = \frac{\sigma^2}{2\pi} \sum_{k=-\infty}^{\infty} \rho(k)e^{-i\omega k}$  of the errors satisfies (Cox, 1984)

$$H(\omega) \sim \tau\omega^{-(1-\delta)} \text{ as } \omega \rightarrow 0,$$

where  $A \sim B$  denotes  $A$  is asymptotic equivalent to  $B$ . In that case,  $\rho(j)$  is of order  $|j|^{-\delta}$  (Adenstedt, 1974). In case of long-range dependence, the correlation decreases more slowly and regression estimation becomes even harder (Hall et al., 1995b; Opsomer et al., 2001). Here, the decrease is of order  $|j|^{-\delta}$  for  $0 < \delta \leq 1$ . Estimation under long-range dependence has attracted more and more attention in recent years. In many scientific research fields such as astronomy, chemistry, physics and signal processing, the observational errors sometimes reveal long-range dependence. Künsch et al. (1993) made the following interesting remark:

*“Perhaps most unbelievable to many is the observation that high-quality measurements series from astronomy, physics, chemistry, generally*

*regarded as prototype of i.i.d. observations, are not independent but long-range correlated.”*

Further, since Kulkarni et al. (2002) have proven consistency for the data-dependent kernel estimators i.e. correlated errors and/or correlation among the independent variables, there is no need to alter the kernel smoother by adding constraints. Confirming their results, we show that the problem is due to the model selection criterion. In fact, we will show in Section 6.3 that there exists a simple multiplicative relation between the bandwidth under correlation and the bandwidth under the i.i.d. assumption.

In the parametric case, ordinary least squares estimators in the presence of autocorrelation are still linear-unbiased as well as consistent, but they are no longer efficient (i.e. minimum variance). As a result, the usual confidence intervals and the test hypotheses cannot be legitimately applied (Sen and Srivastava, 1990).

## 6.2 Problems with Correlation

Some quite fundamental problems occur when nonparametric regression is attempted in the presence of correlated errors. For all nonparametric regression techniques, the shape and the smoothness of the estimated function depends on a large extent on the specific value(s) chosen for the kernel bandwidth (and/or regularization parameter). In order to avoid selecting values for these parameters by trial and error, several data-driven methods are developed (see Chapter 3). However, the presence of correlation between the errors, if ignored, causes breakdown of commonly used automatic tuning parameter selection methods such as CV or plug-in.

Data-driven bandwidth selectors tend to be “fooled” by the correlation, interpreting it as reflecting the regression relationship and variance function. So, the cyclical pattern in positively correlated errors is viewed as a high frequency regression relationship with small variance, and the bandwidth is set small enough to track the cycles resulting in an undersmoothed fitted regression curve. The alternating pattern above and below the true underlying function for negatively correlated errors is interpreted as a high variance, and the bandwidth is set high enough to smooth over the variability, producing an oversmoothed fitted regression curve.

The breakdown of automated methods, as well as a suitable solution, is illustrated by means of a simple example shown in Figure 6.1. For 200 equally spaced observations and a polynomial mean function  $m(x) = 300x^3(1-x)^3$ , four progressively more correlated sets of errors were generated from the same vector of independent noise and added to the mean function. The errors are normally distributed with variance  $\sigma^2 = 0.3$  and correlation following an Auto Regressive

process of order 1, denoted by AR(1),  $\text{corr}(e_i, e_j) = \exp(-\alpha|x_i - x_j|)$  (Fan and Yao, 2003). Figure 6.1 shows four local linear regression estimates (see Chapter 2) for these data sets. For each data set, two bandwidth selection methods were used: standard CV and a correlation-corrected CV (CC-CV) which is further discussed in Section 6.3. Table 6.1 summarizes the bandwidths selected for the four data sets under both methods.

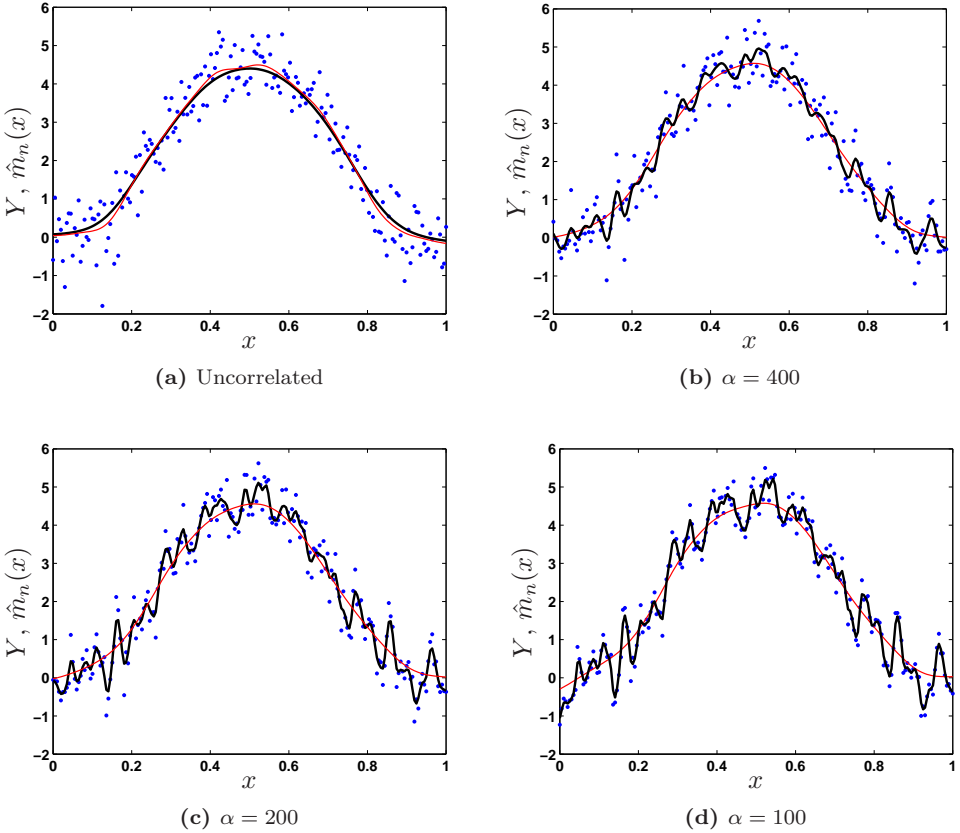
Table 6.1 and Figure 6.1 clearly show that when correlation increases, the bandwidth selected by CV becomes smaller and smaller, and the estimates become more undersmoothed. The bandwidths selected by CC-CV (explained in Section 6.3), a method that accounts for the presence of correlation, are much more stable and result in virtually the same estimate for all four cases. This type of undersmoothing behavior in the presence of positively correlated errors has been observed with most commonly used automated bandwidth selection methods (Altman, 1990; Hart, 1991; Opsomer et al., 2001; Kim et al., 2009).

**Table 6.1:** Summary of bandwidth selection for simulated data in Figure 6.1

Correlation level	Autocorrelation	CV	CC-CV
Independent	0	0.09	0.09
$\alpha = 400$	0.14	0.034	0.12
$\alpha = 200$	0.37	0.0084	0.13
$\alpha = 100$	0.61	0.0072	0.13

## 6.3 New Developments in Kernel Regression with Correlated Errors

In this Section, we address how to deal with, in a simple but effective way, correlated errors using CV. We make a clear distinction between kernel methods requiring no positive definite kernel and kernel methods requiring a positive definite kernel. We will also show that the form of the kernel, based on the mean squared error, is very important when errors are correlated. This is in contrast with the i.i.d. case where the choice between the various kernels, based on the mean squared error, is not very crucial (Härdle, 1999). In what follows, the kernel  $K$  is expected to be an isotropic kernel.



**Figure 6.1:** Simulated data with four levels of AR(1) correlation, estimated with local linear regression; bold line represents estimate obtained with bandwidth selected by CV; thin line represents estimate obtained with bandwidth selected by our method.

### 6.3.1 No Positive Definite Kernel Constraint

To estimate the unknown regression function  $m$ , consider the Nadaraya-Watson (NW) kernel estimator defined as (see also Chapter 2)

$$\hat{m}_n(x) = \sum_{i=1}^n \frac{K\left(\frac{x-x_i}{h}\right)Y_i}{\sum_{j=1}^n K\left(\frac{x-x_j}{h}\right)},$$

where  $h$  is the bandwidth of the kernel  $K$ . This kernel can be one of the following kernels: Epanechnikov, Gaussian, triangular, spline,... An optimal  $h$  can for example be found by minimizing the leave-one-out cross-validation (LOO-CV)

score function

$$\text{LOO-CV}(h) = \frac{1}{n} \sum_{i=1}^n \left( Y_i - \hat{m}_n^{(-i)}(x_i; h) \right)^2, \quad (6.2)$$

where  $\hat{m}_n^{(-i)}(x_i; h)$  denotes the leave-one-out estimator where point  $i$  is left out from the training. For notational ease, the dependence on the bandwidth  $h$  will be suppressed. We can now state the following.

**Lemma 6.1** *Assume the errors are zero-mean, then the expected value of the LOO-CV score function (6.2) is given by*

$$\mathbf{E}[\text{LOO-CV}(h)] = \frac{1}{n} \mathbf{E} \left[ \sum_{i=1}^n \left( m(x_i) - \hat{m}_n^{(-i)}(x_i) \right)^2 \right] + \sigma^2 - \frac{2}{n} \sum_{i=1}^n \mathbf{Cov} \left[ \hat{m}_n^{(-i)}(x_i), e_i \right]$$

PROOF. We first rewrite the LOO-CV score function in a more workable form. Since  $Y_i = m(x_i) + e_i$

$$\begin{aligned} \text{LOO-CV}(h) &= \frac{1}{n} \sum_{i=1}^n [Y_i - \hat{m}_{(-i)}(x_i)]^2 \\ &= \frac{1}{n} \sum_{i=1}^n \left[ m^2(x_i) + 2m(x_i)e_i + e_i^2 - 2Y_i\hat{m}_n^{(-i)}(x_i) + \left( \hat{m}_n^{(-i)}(x_i) \right)^2 \right] \\ &= \frac{1}{n} \sum_{i=1}^n \left[ m(x_i) - \hat{m}_n^{(-i)}(x_i) \right]^2 + \frac{1}{n} \sum_{i=1}^n e_i^2 \\ &\quad + \frac{2}{n} \sum_{i=1}^n \left[ m(x_i) - \hat{m}_n^{(-i)}(x_i) \right] e_i. \end{aligned}$$

Taking expectations yields,

$$\mathbf{E}[\text{LOO-CV}(h)] = \frac{1}{n} \mathbf{E} \left[ \sum_{i=1}^n \left( m(x_i) - \hat{m}_n^{(-i)}(x_i) \right)^2 \right] + \sigma^2 - \frac{2}{n} \sum_{i=1}^n \mathbf{Cov} \left[ \hat{m}_n^{(-i)}(x_i), e_i \right].$$

□

Note that the last term on the right-hand side in Lemma 6.1 is in addition to the correlation already included in the first term. Hart (1991) shows, if  $n \rightarrow \infty$ ,  $nh \rightarrow \infty$ ,  $nh^5 \rightarrow 0$  and for positively correlated errors, that  $\mathbf{E}[\text{LOO-CV}(h)] \approx \sigma^2 + c/nh$  where  $c < 0$  and  $c$  does not depend on the bandwidth. If the correlation is sufficiently strong and  $n$  sufficiently large,  $\mathbf{E}[\text{LOO-CV}(h)]$  will be minimized at a value of  $h$  that is very near to zero. The latter corresponds to almost



interpolating the data (see Figure 6.1). This result does not only hold for leave-one-out cross-validation but also for Mallows’s criterion (Chiu, 1989) and plug-in based techniques (Opsomer et al., 2001). The following theorem provides a simple but effective way to deal with correlated errors. In what follows we will use the following notation

$$k(u) = \int_{-\infty}^{\infty} K(y)e^{-iuy} dy$$

for the Fourier Transform of the kernel function  $K$ .

**Theorem 6.1** *Assume uniform equally spaced design,  $x \in [0,1]$ ,  $\mathbf{E}[e] = 0$ ,  $\mathbf{Cov}[e_i, e_{i+k}] = \mathbf{E}[e_i e_{i+k}] = \gamma_k$  and  $\gamma_k \sim k^{-a}$  for some  $a > 2$ . Assume that*

- (C1)  $K$  is Lipschitz continuous at  $x = 0$ ;
- (C2)  $\int K(u) du = 1, \lim_{|u| \rightarrow \infty} |uK(u)| = 0, \int |K(u)| du < \infty, \sup_u |K(u)| < \infty$ ;
- (C3)  $\int |k(u)| du < \infty$  and  $K$  is symmetric.

Assume further that boundary effects are ignored and that  $h \rightarrow 0$  as  $n \rightarrow \infty$  such that  $nh^2 \rightarrow \infty$ , then for the NW smoother it follows that

$$\begin{aligned} \mathbf{E}[\text{LOO-CV}(h)] &= \frac{1}{n} \mathbf{E} \left[ \sum_{i=1}^n \left( m(x_i) - \hat{m}_n^{(-i)}(x_i) \right)^2 \right] + \sigma^2 \\ &= \frac{4K(0)}{nh - K(0)} \sum_{k=1}^{\infty} \gamma_k + o(n^{-1}h^{-1}). \end{aligned} \tag{6.3}$$

PROOF. Consider only the last term of the expected LOO-CV (Lemma 6.1), i.e.

$$A(h) = -\frac{2}{n} \sum_{i=1}^n \mathbf{Cov} \left[ \hat{m}_n^{(-i)}(x_i), e_i \right].$$

Plugging in the Nadaraya-Watson kernel smoother for  $\hat{m}_n^{(-i)}(x_i)$  in the term above yields

$$A(h) = -\frac{2}{n} \sum_{i=1}^n \mathbf{Cov} \left[ \sum_{\substack{j=1 \\ j \neq i}}^n \frac{K\left(\frac{x_i - x_j}{h}\right) Y_j}{\sum_{l \neq i}^n K\left(\frac{x_i - x_l}{h}\right)}, e_i \right].$$

By using the linearity of the expectation operator,  $Y_j = m(x_j) + e_j$  and  $\mathbf{E}[e] = 0$  it follows that

$$\begin{aligned} A(h) &= -\frac{2}{n} \sum_{i=1}^n \sum_{j \neq i}^n \mathbf{E} \left[ \frac{K\left(\frac{x_i - x_j}{h}\right) Y_j}{\sum_{j \neq i}^n K\left(\frac{x_i - x_l}{h}\right)} e_i \right] \\ &= -\frac{2}{n} \sum_{i=1}^n \sum_{j \neq i}^n \frac{K\left(\frac{x_i - x_j}{h}\right)}{\sum_{j \neq i}^n K\left(\frac{x_i - x_l}{h}\right)} \mathbf{E}[e_i e_j]. \end{aligned}$$

By slightly rewriting the denominator and using the covariance stationary property of the errors (see Definition 6.1), the above equation can be written as

$$A(h) = -\frac{2}{n} \sum_{i=1}^n \sum_{j \neq i}^n \frac{K\left(\frac{x_i - x_j}{h}\right)}{\sum_{j=1}^n K\left(\frac{x_i - x_l}{h}\right) - K(0)} \gamma_{|i-j|}. \quad (6.4)$$

Let  $f$  denote the design density. The first term of the denominator can be written as

$$\begin{aligned} \sum_{j=1}^n K\left(\frac{x_i - x_l}{h}\right) &= nh \hat{f}(x_i) \\ &= nhf(x_i) + nh(\hat{f}(x_i) - f(x_i)). \end{aligned}$$

If conditions (C2) and (C3) are fulfilled,  $f$  is uniform continuous and  $h \rightarrow \infty$  as  $n \rightarrow \infty$  such that  $nh^2 \rightarrow \infty$ , then

$$|\hat{f}(x_i) - f(x_i)| \leq \sup_{x_i} |\hat{f}(x_i) - f(x_i)| \xrightarrow{P} 0 \quad \text{as } n \rightarrow \infty,$$

due to the uniform weak consistency of the kernel density estimator (Parzen, 1962).  $\xrightarrow{P}$  denotes convergence in probability. Hence, for  $n \rightarrow \infty$ , the following approximation is valid

$$nh \hat{f}(x_i) \approx nhf(x_i).$$

Further, by grouping terms together and using the fact that  $x_i \equiv i/n$  (uniform equispaced design) and assume without loss of generality that  $x \in [0,1]$ , (6.4) can be written as

$$\begin{aligned} A(h) &= -\frac{2}{n} \sum_{i=1}^n \frac{1}{nhf(x_i) - K(0)} \sum_{j \neq i}^n K\left(\frac{x_i - x_j}{h}\right) \gamma_{|i-j|} \\ &= -\frac{4}{nh - K(0)} \sum_{k=1}^{n-1} \left(\frac{n-k}{n}\right) K\left(\frac{k}{nh}\right) \gamma_k. \end{aligned}$$

Next, we show that  $\sum_{k=1}^{n-1} \left(\frac{n-k}{n}\right) K\left(\frac{k}{nh}\right) \gamma_k = K(0) \sum_{k=1}^{\infty} \gamma_k + o(n^{-1}h^{-1})$  for  $n \rightarrow \infty$ . Since the kernel  $K \geq 0$  is Lipschitz continuous at  $x = 0$

$$[K(0) + C_2x]_+ \leq K(x) \leq K(0) + C_1x,$$

where  $[z]_+ = \max(z, 0)$ . Then, for  $K(0) \geq 0$  and  $C_1 > C_2$ , we establish the following upperbound

$$\begin{aligned} \sum_{k=1}^{n-1} \left(\frac{n-k}{n}\right) K\left(\frac{k}{nh}\right) \gamma_k &\leq \sum_{k=1}^{n-1} \left(1 - \frac{k}{n}\right) \left(K(0) + C_1 \frac{k}{nh}\right) \gamma_k \\ &\leq \sum_{k=1}^{n-1} K(0) \gamma_k + \sum_{k=1}^{n-1} C_1 \frac{k}{nh} \gamma_k. \end{aligned}$$

Then, for  $n \rightarrow \infty$  and using  $\gamma_k \sim k^{-a}$  for  $a > 2$ ,

$$C_1 \sum_{k=1}^{n-1} \frac{k}{nh} \gamma_k = C_1 \sum_{k=1}^{n-1} \frac{k^{1-a}}{nh} = o(n^{-1}h^{-1}).$$

Hence,

$$\sum_{k=1}^{n-1} \left(\frac{n-k}{n}\right) K\left(\frac{k}{nh}\right) \gamma_k \leq K(0) \sum_{k=1}^{\infty} \gamma_k + o(n^{-1}h^{-1}).$$

For the construction of the lower bound, assume first that  $C_2 < 0$  and  $K(0) \geq 0$  then

$$\sum_{k=1}^{n-1} \left(\frac{n-k}{n}\right) K\left(\frac{k}{nh}\right) \gamma_k \geq \sum_{k=1}^{n-1} \left(1 - \frac{k}{n}\right) \left[K(0) + C_2 \frac{k}{nh}\right]_+ \gamma_k.$$

Since  $C_2 < 0$ , it follows that  $k \leq \frac{K(0)}{-C_2}nh$  and therefore

$$\sum_{k=1}^{n-1} \left(1 - \frac{k}{n}\right) \left[K(0) + C_2 \frac{k}{nh}\right]_+ \gamma_k = \sum_{k=1}^{\min(n-1, \frac{K(0)}{-C_2}nh)} \left(1 - \frac{k}{n}\right) \left(K(0) + C_2 \frac{k}{nh}\right) \gamma_k.$$

Analogous to deriving the upper bound, we obtain for  $n \rightarrow \infty$

$$\sum_{k=1}^{n-1} \left(\frac{n-k}{n}\right) K\left(\frac{k}{nh}\right) \gamma_k \geq K(0) \sum_{k=1}^{\infty} \gamma_k + o(n^{-1}h^{-1}).$$

In the second case i.e.  $C_2 > 0$ , the same lower bound can be obtained. Finally, from the upper and lower bound, for  $n \rightarrow \infty$ , yields

$$\sum_{k=1}^{n-1} \left(\frac{n-k}{n}\right) K\left(\frac{k}{nh}\right) \gamma_k = K(0) \sum_{k=1}^{\infty} \gamma_k + o(n^{-1}h^{-1}). \quad \square$$

From this result it is clear that, by taking a kernel satisfying the condition  $K(0) = 0$ , the correlation structure is removed without requiring any prior information about its structure and (6.3) reduces to

$$\mathbf{E}[\text{LOO-CV}(h)] = \frac{1}{n} \mathbf{E} \left[ \sum_{i=1}^n \left( m(x_i) - \hat{m}_n^{(-i)}(x_i) \right)^2 \right] + \sigma^2 + o(n^{-1}h^{-1}). \quad (6.5)$$

Therefore, it is natural to use a bandwidth selection criterion based on a kernel satisfying  $K(0) = 0$ , defined by

$$\hat{h}_b = \arg \min_{h \in \mathcal{Q}_n} \text{LOO-CV}(h),$$

where  $\mathcal{Q}_n$  is a finite set of parameters. We have relaxed the conditions of Kim et al. (2009) in order to derive Theorem 6.1, i.e. they require the kernel to be differentiable at zero while in our proof the kernel only needs to satisfy a Lipschitz condition at zero. Further, we extended the proof for the NW estimator. As will be shown later, this relaxation can lead to kernel classes resulting in a better regression estimate (lower mean squared error).

Notice that if  $K$  is a symmetric probability density function, then  $K(0) = 0$  implies that  $K$  is not unimodal. Hence, it is obvious to use bimodal kernels. Such a kernel gives more weight to observations near to the point  $x$  of interest than those that are far from  $x$ . But at the same time it also reduces the weight of points which are too close to  $x$ . A major advantage of using a bandwidth selection criterion based on bimodal kernels is the fact that is more efficient in removing the correlation than leave- $(2l+1)$ -out CV (Chu and Marron, 1991b).

**Definition 6.2 (Leave- $(2l+1)$ -out CV)** *Leave- $(2l+1)$ -out CV or modified CV (MCV) is defined as*

$$\text{MCV}(h) = \frac{1}{n} \sum_{i=1}^n \left( Y_i - \hat{m}_n^{(-i)}(x_i) \right)^2, \quad (6.6)$$

where  $\hat{m}_n^{(-i)}(x_i)$  is the leave- $(2l+1)$ -out version of  $m(x_i)$ , i.e. the observations  $(x_{i+j}, Y_{i+j})$  for  $-l \leq j \leq l$  are left out to estimate  $\hat{m}_n(x_i)$ .

Taking a bimodal kernel satisfying  $K(0) = 0$  results in (6.5) while leave- $(2l+1)$ -out CV with unimodal kernel  $K$ , under the conditions of Theorem 6.1, yields

$$\begin{aligned} \mathbf{E}[\text{MCV}(h)] &= \frac{1}{n} \mathbf{E} \left[ \sum_{i=1}^n \left( m(x_i) - \hat{m}_n^{(-i)}(x_i) \right)^2 \right] + \sigma^2 \\ &\quad - \frac{4K(0)}{nh - K(0)} \sum_{k=l+1}^{\infty} \gamma_k + o(n^{-1}h^{-1}). \end{aligned}$$

The formula above clearly shows that leave- $(2l + 1)$ -out CV with unimodal kernel  $K$  cannot completely remove the correlation structure. Only the first  $l$  elements of the correlation are removed.

Another possibility of bandwidth selection under correlation, not based on bimodal kernels, is to estimate the covariance structure  $\gamma_0, \gamma_1, \dots$  in (6.3). Although the usual residual-based estimators of the autocovariances  $\hat{\gamma}_k$  are consistent,  $\sum_{k=1}^{\infty} \hat{\gamma}_k$  is not a consistent estimator of  $\sum_{k=1}^{\infty} \gamma_k$  (Simonoff, 1996). A first approach correcting for this, is to estimate  $\sum_{k=1}^{\infty} \gamma_k$  by fitting a parametric model to the residuals (and thereby obtaining estimates of  $\gamma_k$ ) and use these estimates in (6.3) together with a univariate kernel. If the assumed parametric model is incorrect, these estimates can be far from the correct ones resulting in a poor choice of the bandwidth. However, Altman (1990) showed that, if the signal to noise ratio is small, this approach results in sufficiently good estimates of correlation for correcting the selection criteria. A second approach, proposed by Hart (1989, 1991), suggests estimating the covariance structure in the spectral domain via differencing the data at least twice. A third approach is to derive an asymptotic bias-variance decomposition under the correlated error assumption of the kernel smoother. In this way and under certain conditions on the correlation function, plug-ins can be derived taking the correlation into account, see e.g. Hermann et al. (1992), Opsomer et al. (2001), Hall and Keilegom (2003), Francisco-Fernández and Opsomer (2004) and Francisco-Fernández et al. (2005). More recently, Park et al. (2006) proposed to estimate the error correlation nonparametrically without prior knowledge of the correlation structure.

### 6.3.2 Positive Definite Kernel Constraint

From Chapter 2, we know that methods like SVM and LS-SVM require a positive (semi) definite kernel. However, the following proposition reveals why a bimodal kernel  $\tilde{K}$  cannot be directly applied in these methods.

**Proposition 6.1** *A bimodal kernel  $\tilde{K}$  is never positive (semi) definite.*

PROOF. We split up the proof in two parts, i.e. for positive definite and positive semi-definite kernels. The statement will be proven by contradiction.

- Suppose there exists a positive definite bimodal kernel  $\tilde{K}$ . This leads to a positive definite kernel matrix  $\Omega$ . Then, all eigenvalues of  $\Omega$  are strictly positive and hence the trace of  $\Omega$  is always larger than zero. However, this is in contradiction with the fact that  $\Omega$  has all zeros on its main diagonal. Consequently, a positive definite bimodal kernel  $\tilde{K}$  cannot exist.

- Suppose there exists a positive semi-definite bimodal kernel  $\tilde{K}$ . Then, at least one eigenvalue of the matrix  $\Omega$  is equal to zero (the rest of the eigenvalues is strictly positive). We have now two possibilities i.e. some eigenvalues are equal to zero and all eigenvalues are equal to zero. In the first case, the trace of the matrix  $\Omega$  is larger than zero and we have again a contradiction. In the second case, the trace of the matrix  $\Omega$  is equal to zero and also the determinant of  $\Omega$  equals zero (since all eigenvalues are equal to zero). But the determinant can never be zero since there is no linear dependence between the rows or columns (there is a zero in each row or column). This concludes the proof. □

Consequently, the previous strategy of using bimodal kernels cannot directly be applied to SVM and LS-SVM. A possible way to circumvent this obstacle, is to use the bandwidth  $\hat{h}_b$ , obtained from the bimodal kernel, as a pilot bandwidth selector for other data-driven selection procedures such as leave- $(2l + 1)$ -out CV or block bootstrap bandwidth selector (Hall et al., 1995b). Since the block bootstrap in Hall et al. (1995b) is based on two smoothers, i.e. one is used to compute centered residuals and the other generates bootstrap data, the procedure is computationally costly. Therefore, we will use leave- $(2l + 1)$ -out CV or MCV which has a lower computational cost. A crucial parameter to be estimated in MCV, see also Chu and Marron (1991b), is  $l$ . Indeed, the amount of dependence between  $\hat{m}_n(x_k)$  and  $Y_k$  is reduced as  $l$  increases.

A similar problem arises in block bootstrap where the accuracy of the method critically depends on the block size that is supplied by the user. The orders of magnitude of the optimal block sizes are known in some inference problems (see Künsch, 1989; Hall et al., 1995a; Lahiri, 1999; Bühlmann and Künsch, 1999). However, the leading terms of these optimal block sizes depend on various population characteristics in an intricate manner, making it difficult to estimate these parameters in practice. Recently, Lahiri et al. (2007) proposed a nonparametric plug-in principle to determine the block size.

For  $l = 0$ , MCV is ordinary CV or leave-one-out CV. One possible method to select a value for  $l$  is to use  $\hat{h}_b$  as pilot bandwidth selector. Define a bimodal kernel  $\tilde{K}$  and assume  $\hat{h}_b$  is available, then one can calculate

$$\hat{m}_n(x) = \sum_{i=1}^n \frac{\tilde{K}\left(\frac{x-x_i}{\hat{h}_b}\right) Y_i}{\sum_{j=1}^n \tilde{K}\left(\frac{x-x_j}{\hat{h}_b}\right)}. \quad (6.7)$$

From this result, the residuals are obtained by

$$\hat{e}_i = Y_i - \hat{m}_n(x_i), \text{ for } i = 1, \dots, n$$

and choose  $l$  to be the smallest  $q \geq 1$  such that

$$|r_q| = \left| \frac{\sum_{i=1}^{n-q} \hat{e}_i \hat{e}_{i+q}}{\sum_{i=1}^n \hat{e}_i^2} \right| \leq \frac{\Phi^{-1}(1 - \frac{\alpha}{2})}{\sqrt{n}}, \tag{6.8}$$

where  $\Phi^{-1}$  denotes the quantile function of the standard normal distribution and  $\alpha$  is the significance level, say 5%. Observe that (6.8) is based on the fact that  $r_q$  is asymptotically normal distributed under the centered i.i.d. error assumption (Kendall et al., 1983) and hence provides an approximate  $100(1 - \alpha)\%$  confidence interval for the autocorrelation. The reason why (6.8) can be legitimately applied is motivated by combining the theoretical results of Kim et al. (2004) and Park et al. (2006) stating that

$$\frac{1}{n - q} \sum_{i=1}^{n-q} \hat{e}_i \hat{e}_{i+q} = \frac{1}{n - q} \sum_{i=1}^{n-q} e_i e_{i+q} + O(n^{-4/5}).$$

Once  $l$  is selected, the tuning parameters of SVM or LS-SVM can be determined by using leave- $(2l + 1)$ -out CV combined with a positive definite kernel, e.g. Gaussian kernel. We then call Correlation-Corrected CV (CC-CV) the combination of finding  $l$  via bimodal kernels and using the obtained  $l$  in leave- $(2l + 1)$ -out CV. Algorithm 6 summarizes the CC-CV procedure for LS-SVM.

---

**Algorithm 6** Correlation-Corrected CV for LS-SVM

---

- 1: Determine  $\hat{h}_b$  in (6.7) with a bimodal kernel by means of LOO-CV
  - 2: Calculate  $l$  satisfying (6.8)
  - 3: Determine both tuning parameters for LS-SVM by means of leave- $(2l + 1)$ -out CV (6.6) and a positive definite unimodal kernel.
- 

### 6.3.3 Drawback of Using Bimodal Kernels

Although bimodal kernels are very effective in removing the correlation structure, they have an inherent drawback. When using bimodal kernels to estimate the regression function  $m$ , the estimate  $\hat{m}_n$  will suffer from increased mean squared error (MSE). The following theorem indicates the asymptotic behavior of the MSE of  $\hat{m}_n(x)$  when the errors are covariant stationary.

**Theorem 6.2 (Simonoff, 1996)** *Let (6.1) hold and assume that  $m$  has two continuous derivatives. Assume also that  $\mathbf{Cov}[e_i, e_{i+k}] = \gamma_k$  for all  $k$ , where  $\gamma_0 = \sigma^2 < \infty$  and  $\sum_{k=1}^{\infty} k|\gamma_k| < \infty$ . Now, as  $n \rightarrow \infty$  and  $h \rightarrow 0$ , the following statement holds uniformly in  $x \in (h, 1 - h)$  for the Mean Integrated*

*Squared Error (MISE)*

$$\text{MISE}(\hat{m}_n) = \frac{\mu_2^2(K)h^4 \int (m''(x))^2 dx}{4} + \frac{R(K)[\sigma^2 + 2 \sum_{k=1}^{\infty} \gamma_k]}{nh} + o(h^4 + n^{-1}h^{-1}),$$

where  $\mu_2(K) = \int u^2 K(u) du$  and  $R(K) = \int K^2(u) du$ .

An asymptotic optimal constant or global bandwidth  $\hat{h}_{\text{AMISE}}$ , for  $m''(x) \neq 0$ , is the minimizer of the Asymptotic MISE (AMISE)

$$\text{AMISE}(\hat{m}_n) = \frac{\mu_2^2(K)h^4 \int (m''(x))^2 dx}{4} + \frac{R(K)[\sigma^2 + 2 \sum_{k=1}^{\infty} \gamma_k]}{nh},$$

w.r.t. to the bandwidth, yielding

$$\hat{h}_{\text{AMISE}} = \left[ \frac{R(K)[\sigma^2 + 2 \sum_{k=1}^{\infty} \gamma_k]}{\mu_2^2(K) \int (m''(x))^2 dx} \right]^{1/5} n^{-1/5}. \quad (6.9)$$

We see that  $\hat{h}_{\text{AMISE}}$  is at least as big as the bandwidth for i.i.d data  $\hat{h}_0$  if  $\gamma_k \geq 0$  for all  $k \geq 1$ . The following corollary shows that there is a simple multiplicative relationship between the asymptotic optimal bandwidth for dependent data  $\hat{h}_{\text{AMISE}}$  and bandwidth for independent data  $\hat{h}_0$ .

**Corollary 6.1** *Assume the conditions of Theorem 6.2 hold, then*

$$\hat{h}_{\text{AMISE}} = \left[ 1 + 2 \sum_{k=1}^{\infty} \rho(k) \right]^{1/5} \hat{h}_0, \quad (6.10)$$

where  $\hat{h}_{\text{AMISE}}$  is the asymptotic MISE optimal bandwidth for dependent data,  $\hat{h}_0$  is the asymptotic optimal bandwidth for independent data and  $\rho(k)$  denotes the autocorrelation function at lag  $k$ , i.e.  $\rho(k) = \gamma_k/\sigma^2 = \mathbf{E}[e_i e_{i+k}]/\sigma^2$ .

PROOF. From (6.9) it follows that

$$\begin{aligned} \hat{h}_{\text{AMISE}} &= \left[ \frac{R(K)\sigma^2}{n\mu_2^2(K) \int (m''(x))^2 dx} + \frac{2R(K) \sum_{k=1}^{\infty} \gamma_k}{n\mu_2^2(K) \int (m''(x))^2 dx} \right]^{1/5} \\ &= \left[ \hat{h}_0^5 + \frac{\sigma^2 R(K)}{n\mu_2^2(K) \int (m''(x))^2 dx} \frac{2 \sum_{k=1}^{\infty} \gamma_k}{\sigma^2} \right]^{1/5} \\ &= \left[ 1 + 2 \sum_{k=1}^{\infty} \rho(k) \right]^{1/5} \hat{h}_0. \end{aligned}$$

□



Thus, if the data are positively autocorrelated ( $\rho(k) \geq 0 \ \forall k$ ), the optimal bandwidth under correlation is larger than that for independent data. Unfortunately, (6.10) is quite hard to use in practice since it requires knowledge about the correlation structure and an estimate of the bandwidth  $\hat{h}_0$  under the i.i.d. assumption, given correlated data.

By taking  $\hat{h}_{AMISE}$  as in (6.9), the corresponding asymptotic MISE is equal to

$$AMISE(\hat{m}_n) = cD_K^{2/5} n^{-4/5},$$

where  $c$  depends neither on the bandwidth nor on the kernel  $K$  and

$$D_K = \mu_2(K)R(K)^2 = \left( \int u^2 K(u) du \right) \left( \int K^2(u) du \right)^2. \tag{6.11}$$

It is obvious that one wants to minimize (6.11) with respect to the kernel function  $K$ . This leads to the well-known Epanechnikov kernel  $K_{\text{epa}}$ . However, adding the constraint  $K(0) = 0$  (see Theorem 6.1) to the minimization of (6.11) would lead to the following optimal kernel

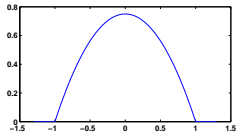
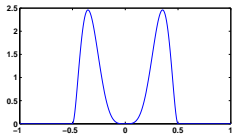
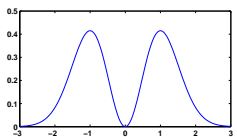
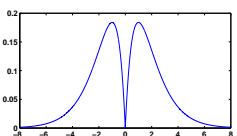
$$K^*(u) = \begin{cases} K_{\text{epa}}(u), & \text{if } u \neq 0; \\ 0, & \text{if } u = 0. \end{cases}$$

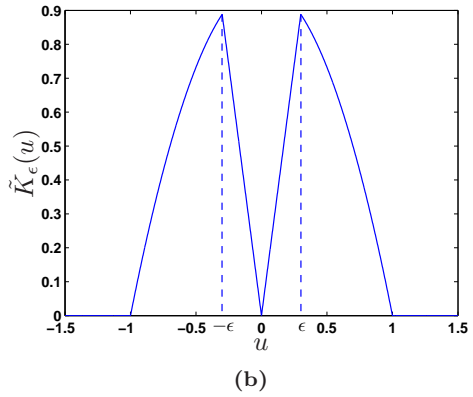
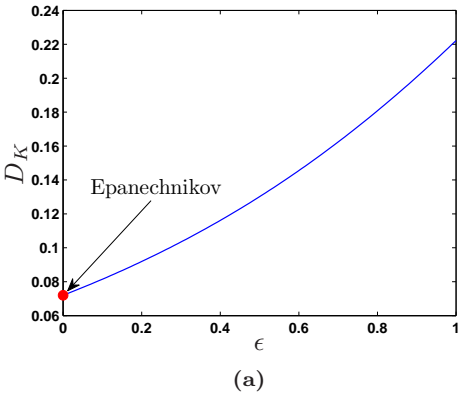
Certainly, this kernel violates assumption (C1) in Theorem 6.1. In fact, an optimal kernel does not exist in the class of kernels satisfying assumption (C1) and  $K(0) = 0$ . To illustrate this, note that there exist a sequence of kernels  $\{K_{\text{epa}}(u, \epsilon)\}_{\epsilon \in ]0, 1[}$ , indexed by  $\epsilon$ , such that  $K_{\text{epa}}(u)$  converges to  $K^*(u)$  and the value of  $\int K_{\text{epa}}(u, \epsilon)^2 du$  decreases to  $\int K^*(u)^2 du$  as  $\epsilon$  tends to zero. Since an optimal kernel in this class cannot be found, we have to be content with a so-called  $\epsilon$ -optimal class of bimodal kernels  $\tilde{K}_\epsilon(u)$ , with  $0 < \epsilon < 1$ , defined as

$$\tilde{K}_\epsilon(u) = \frac{4}{4 - 3\epsilon - \epsilon^3} \begin{cases} \frac{3}{4}(1 - u^2)I_{\{|u| \leq 1\}}, & |u| \geq \epsilon; \\ \frac{3}{4} \frac{1 - \epsilon^2}{\epsilon} |u|, & |u| < \epsilon. \end{cases} \tag{6.12}$$

For  $\epsilon = 0$ , we define  $\tilde{K}_\epsilon(u) = K_{\text{epa}}(u)$ . Table 6.2 displays several possible bimodal kernel functions with their respective  $D_K$  value compared to the Epanechnikov kernel. Although it is possible to express the  $D_K$  value for  $\tilde{K}_\epsilon(u)$  as a function of  $\epsilon$ , we do not include it in Table 6.2 but instead, we graphically illustrate the dependence of  $D_K$  on  $\epsilon$  in Figure 6.2a. An illustration of the  $\epsilon$ -optimal class of bimodal kernels is shown in Figure 6.2b.

**Table 6.2:** Kernel functions with illustrations and their respective  $D_K$  value compared to the Epanechnikov kernel.  $I_A$  denotes the indicator function of an event  $A$ .

	kernel function	Illustration	$D_K$
$K_{\text{epa}}$	$\frac{3}{4}(1 - u^2)I_{\{ u  \leq 1\}}$		0.072
$\tilde{K}_1$	$630(4u^2 - 1)^2 u^4 I_{\{ u  \leq 1/2\}}$		0.374
$\tilde{K}_2$	$\frac{2}{\sqrt{\pi}} u^2 \exp(-u^2)$		0.134
$\tilde{K}_3$	$\frac{1}{2} u  \exp(- u )$		0.093



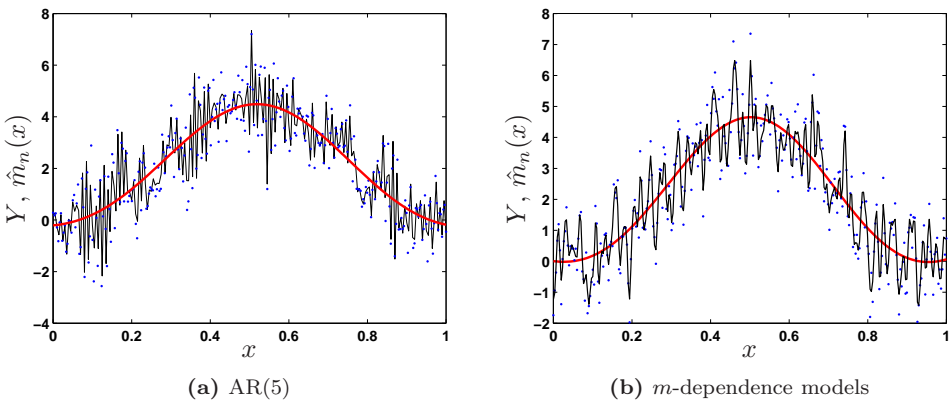
**Figure 6.2:** (a)  $D_K$  as a function of  $\epsilon$  for the  $\epsilon$ -optimal class of kernels. The dot on the left side marks the Epanechnikov kernel; (b) Illustration of the  $\epsilon$ -optimal class of kernels for  $\epsilon = 0.3$ .

## 6.4 Simulations

### 6.4.1 CC-CV vs. LOO-CV with Different Noise Models

In a first example, we compare the finite sample performance of CC-CV with the classical leave-one-out CV (LOO-CV) based on a unimodal kernel in the presence of correlation. Consider the following function  $m(x) = 300x^3(1-x)^3$  for  $0 \leq x \leq 1$ . The sample size is set to  $n = 200$ . We consider two types of noise models: (i) an AR(5) process  $e_j = \sum_{l=1}^5 \phi_l e_{j-l} + \sqrt{1-\phi_1^2} Z_j$  where  $Z_j$  are i.i.d. normal random variables with variance  $\sigma^2 = 0.5$  and zero mean. The errors  $e_j$  for  $j = 1, \dots, 5$  are standard normal random variables. The AR(5) parameters are set to  $[\phi_1, \phi_2, \phi_3, \phi_4, \phi_5] = [0.7, -0.5, 0.4, -0.3, 0.2]$ . (ii)  $m$ -dependent models  $e_i = r_0 \delta_i + r_1 \delta_{i-1}$  with  $m = 1$  where  $\delta_i$  is i.i.d. standard normal random variable,  $r_0 = \frac{\sqrt{1+2\nu} + \sqrt{1-2\nu}}{2}$  and  $r_1 = \frac{\sqrt{1+2\nu} - \sqrt{1-2\nu}}{2}$  for  $\nu = 1/2$ .

Figure 6.3 shows typical results of the regression estimates for both noise models. Table 6.3 summarizes the average of the regularization parameters, bandwidths and asymptotic squared error, defined as  $\text{ASE} = \frac{1}{n} \sum_{i=1}^n (m(x_i) - \hat{m}_n(x_i))^2$ , for 50 runs for both noise models. By looking at the average ASE, it is clear that the tuning parameters obtained by CC-CV result into better estimates which are not influenced by the correlation. Also notice the small bandwidths and larger regularization constants found by LOO-CV for both noise models. This provides clear evidence that the kernel smoother is trying to model the noise instead of the true underlying function. These findings are also valid if one uses generalized CV or  $v$ -fold CV. Figure 6.4 and Figure 6.5 show the CV surfaces for both model

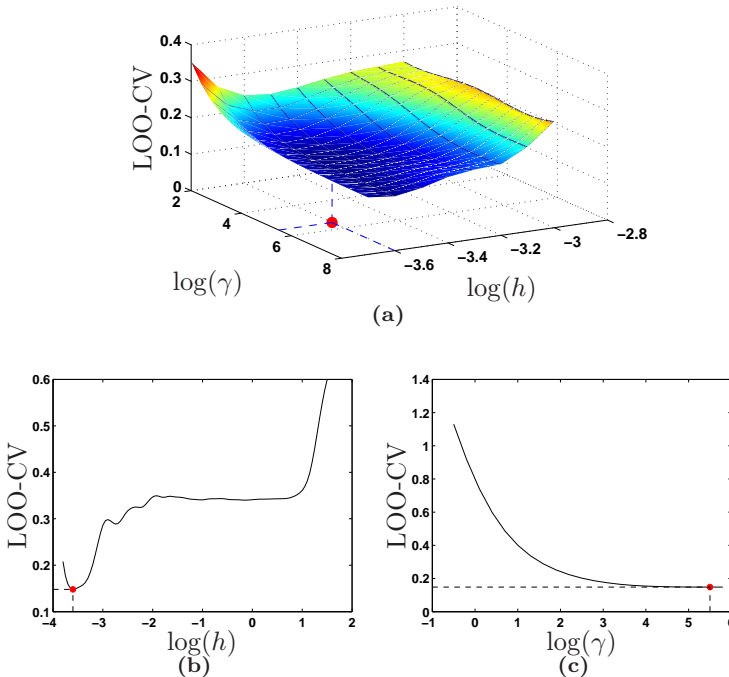


**Figure 6.3:** Typical results of the LS-SVM regression estimates for both noise models. The thin line represents the estimate with tuning parameters determined by LOO-CV and the bold line is the estimate based on the CC-CV tuning parameters.

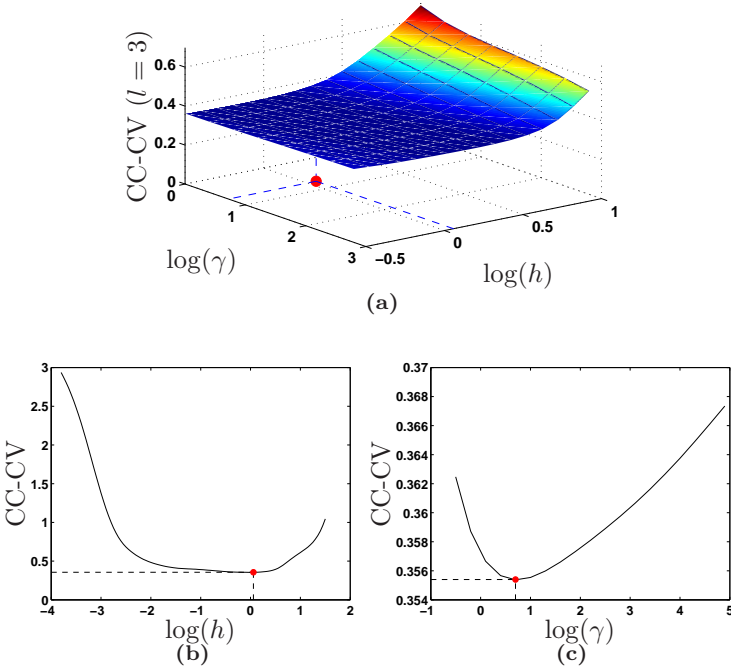
**Table 6.3:** Average of the regularization parameters, bandwidths and average ASE for 50 runs for both noise models

	AR(5)		$m$ -dependence models	
	LOO-CV	CC-CV	LOO-CV	CC-CV
$\hat{\gamma}$	224.69	2.28	$1.03 \times 10^5$	6.96
$\hat{h}$	0.027	1.06	0.03	1.89
av. ASE	0.36	0.021	0.89	0.04

selection methods on the AR(5) noise model. These plots clearly demonstrate the shift of the tuning parameters. A cross section for both tuning parameters is provided below each surface plot. Also note that the surface of the CC-CV tends to be flatter than LOO-CV and so it is harder to minimize numerically (see Hall et al., 1995b). Because of this extra difficulty, we used the optimization approach discussed at the end of Chapter 3 in all the examples.



**Figure 6.4:** (a) CV surface for LOO-CV; (b) cross sectional view of  $\log(h)$  for fixed  $\log(\gamma) = 5.5$ ; (c) cross sectional view of  $\log(\gamma)$  for fixed  $\log(h) = -3.6$ . The dot indicates the minimum of the cost function obtained by Coupled Simulated Annealing with simplex search. These results correspond with the first column of Table 6.3.



**Figure 6.5:** (a) CV surface for CC-CV; (b) cross sectional view of  $\log(h)$  for fixed  $\log(\gamma) = 0.82$ ; (c) cross sectional view of  $\log(\gamma)$  for fixed  $\log(h) = 0.06$ . The dot indicates the minimum of the cost function obtained by Coupled Simulated Annealing with simplex search. These results correspond with the second column of Table 6.3.

### 6.4.2 Evolution of the Bandwidth Under Correlation

Consider the same function as in the previous simulation and let  $n = 400$ . The noise error model is taken to be an AR(1) process with varying parameter  $\phi = -0.95, -0.9, \dots, 0.9, 0.95$ . For each  $\phi$ , 100 replications of size  $n$  were made to report the average regularization parameter, bandwidth and average ASE for both methods. The results are summarized in Table 6.4. The results indicate that the CC-CV method is indeed capable of finding good tuning parameters in the presence of correlated errors. The CC-CV method outperforms the classical LOO-CV for positively correlated errors, i.e.  $\phi > 0$ . The method is capable of producing good bandwidths which do not tend to very small values as in the LOO-CV case. In general, the regularization parameter obtained by LOO-CV is larger than the one from CC-CV. However, the latter is not theoretically verified and serves only as a heuristic.

On the other hand, for negatively correlated errors ( $\phi < 0$ ), both methods perform equally well. The reason why the effects from correlated errors is more outspoken for positive  $\phi$  than for negative  $\phi$  might be related to the fact that negatively correlated errors are seemingly hard to differentiate from i.i.d. errors in practice.

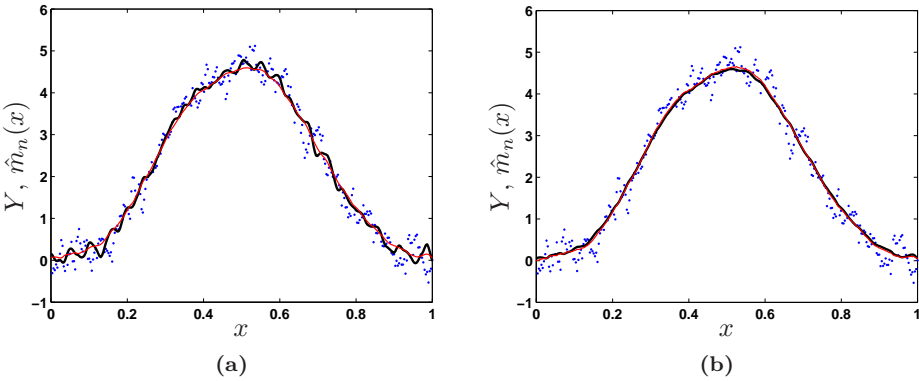
**Table 6.4:** Average of the regularization parameters, bandwidths and average ASE for 50 runs for the AR(1) process with varying parameter  $\phi$

$\phi$	LOO-CV			CC-CV		
	$\hat{\gamma}$	$\hat{h}$	av. ASE	$\hat{\gamma}$	$\hat{h}$	av. ASE
-0.95	14.75	1.48	0.0017	7.65	1.43	0.0019
-0.9	11.48	1.47	0.0017	14.58	1.18	0.0021
-0.8	7.52	1.39	0.0021	18.12	1.15	0.0031
-0.7	2.89	1.51	0.0024	6.23	1.21	0.0030
-0.6	28.78	1.52	0.0030	5.48	1.62	0.0033
-0.5	42.58	1.71	0.0031	87.85	1.75	0.0048
-0.4	39.15	1.55	0.0052	39.02	1.43	0.0060
-0.3	72.91	1.68	0.0055	19.76	1.54	0.0061
-0.2	98.12	1.75	0.0061	99.56	1.96	0.0069
-0.1	60.56	1.81	0.0069	101.1	1.89	0.0070
0	102.5	1.45	0.0091	158.4	1.89	0.0092
0.1	1251	1.22	0.0138	209.2	1.88	0.0105
0.2	1893	0.98	0.0482	224.6	1.65	0.0160
0.3	1535	0.66	0.11	5.18	1.86	0.0161
0.4	482.3	0.12	0.25	667.5	1.68	0.023
0.5	2598	0.04	0.33	541.8	1.82	0.033
0.6	230.1	0.03	0.36	986.9	1.85	0.036
0.7	9785	0.03	0.41	12.58	1.68	0.052
0.8	612.1	0.03	0.45	1531	1.53	0.069
0.9	448.8	0.02	0.51	145.12	1.35	0.095
0.95	878.4	0.01	0.66	96.5	1.19	0.13

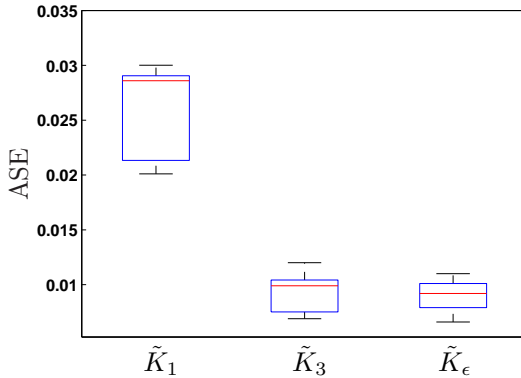
### 6.4.3 Comparison of Different Bimodal Kernels

Consider a polynomial mean function  $m(x_k) = 300x_k^3(1 - x_k)^3$ ,  $k = 1, \dots, 400$ , where the errors are normally distributed with variance  $\sigma^2 = 0.1$  and correlation following an AR(1) process,  $\text{corr}(e_i, e_j) = \exp(-150|x_i - x_j|)$ . The simulation

shows the difference in regression estimates (Nadaraya-Watson) based on kernels  $\tilde{K}_1$ ,  $\tilde{K}_3$  and  $\tilde{K}_\epsilon$  with  $\epsilon = 0.1$ , see Figure 6.6a and 6.6b respectively. Due to the larger  $D_K$  value of  $\tilde{K}_1$ , the estimate tends to be more wiggly compared to kernel  $\tilde{K}_3$ . The difference between the regression estimate based on  $\tilde{K}_3$  and  $\tilde{K}_\epsilon$  with  $\epsilon = 0.1$  is very small and almost cannot be seen on Figure 6.6b. For the sake of comparison between regression estimates based on  $\tilde{K}_1$ ,  $\tilde{K}_3$  and  $\tilde{K}_\epsilon$  with  $\epsilon = 0.1$ , we show the corresponding asymptotic squared error (ASE) in Figure 6.7 based on 100 simulations with the data generation process described as above. The boxplot shows that the kernel  $\tilde{K}_\epsilon$  with  $\epsilon = 0.1$  outperforms the other two.



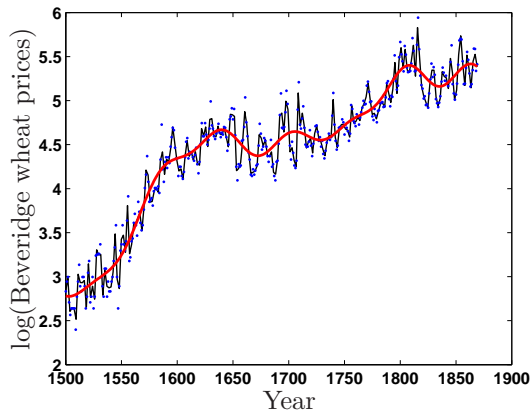
**Figure 6.6:** Difference in the regression estimate (Nadaraya-Watson) (a) based on kernel  $\tilde{K}_1$  (bold line) and  $\tilde{K}_3$  (thin line). Due to the larger  $D_K$  value of  $\tilde{K}_1$ , the estimate tends to be more wiggly compared to  $\tilde{K}_3$ ; (b) based on kernel  $\tilde{K}_3$  (bold line) and  $\epsilon$ -optimal kernel with  $\epsilon = 0.1$  (thin line).



**Figure 6.7:** Boxplot of the asymptotic squared errors for the regression estimates based on bimodal kernels  $\tilde{K}_1$ ,  $\tilde{K}_3$  and  $\tilde{K}_\epsilon$  with  $\epsilon = 0.1$ .

### 6.4.4 Real life data sets

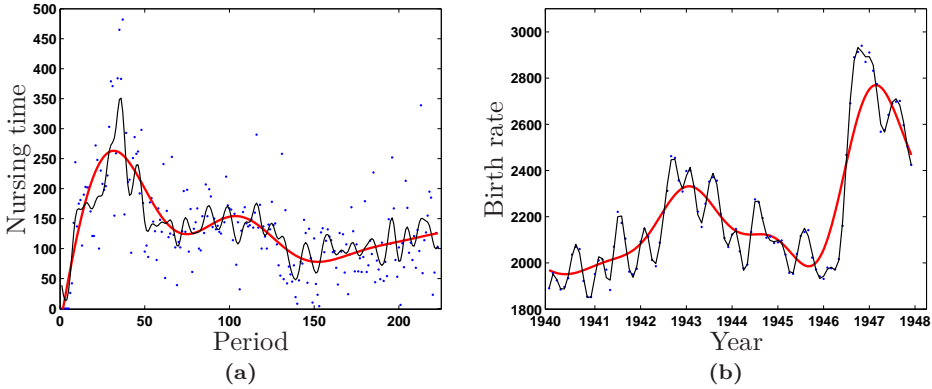
First, we apply the proposed method to a time series of the Beveridge (1921) index of wheat prices from the year 1500 to 1869 (Anderson, 1971). These data are an annual index of prices at which wheat was sold in European markets. The data used for analysis are the natural logarithms of the Beveridge indices. This transformation is done to correct for heteroscedasticity in the original series. The result is shown in Figure 6.8 for LS-SVM with Gaussian kernel. It is clear that the estimate based on classical leave-one-out CV (assumption of no correlation) is very rough. The proposed CC-CV method produces a smooth regression fit.



**Figure 6.8:** Difference in regression estimates (LS-SVM) for standard leave-one-out CV (thin line) and the proposed method (bold line).

As a second and third example, we consider the nursing time of the beluga whale (see Figure 6.9a) and birth rate (see Figure 6.9b) data set (Simonoff, 1996). Figure 6.9a shows the scatter plot that relates the nursing time (in seconds) of a newborn beluga whale calf Hudson to the time after birth, where time is measured in six-hour time periods. Figure 6.9b shows the the U.S. monthly birth rate for the period from January 1940 through December 1947. As before, the proposed CC-CV method produces a smooth regression fit.





**Figure 6.9:** Typical results of the LS-SVM regression estimates for the (a) nursing time of the beluga whale and (b) birth rate data set. The thin line represents the estimate with tuning parameters determined by LOO-CV and the bold line is the estimate based on the CC-CV tuning parameters.

## 6.5 Conclusions

In this Chapter, we investigated the possible consequences when the i.i.d. assumption was violated. We showed that classical model selection procedures break down in the presence of correlation rather than the nonparametric regression methods. Since the latter stays consistent when correlation is present in the data, it is not necessary to modify or add extra constraints to the smoother. In order to cope with the problem of correlation, we proved that by taking a kernel  $K$  satisfying  $K(0) = 0$ , the correlation structure is successfully removed without requiring any prior knowledge about its structure. Further, we showed both theoretically and experimentally, that by using bimodal kernels the estimate will suffer from increased mean squared error. We developed a class of so-called  $\epsilon$ -optimal class of bimodal kernels, since an optimal bimodal kernel satisfying  $K(0) = 0$  cannot be found, which reduces this effect as much as possible. Finally, we proposed, based on our theoretical justifications, a model selection procedure (CC-CV) for LS-SVM in order to effectively handle correlation in the data.



# Chapter 7

## Confidence and Prediction Intervals

In this Chapter, we discuss the construction of bias-corrected  $100(1 - \alpha)\%$  approximate confidence and prediction intervals (pointwise and uniform) for linear smoothers, in particular for LS-SVM. We prove the asymptotic normality of LS-SVM. Also, we discuss a technique called double smoothing to determine the bias without estimating higher order derivatives. Further, we develop a nonparametric variance estimator which can be related to other well-known nonparametric variance estimators. In order to obtain uniform or simultaneous confidence intervals, we will use two techniques: (i) Bonferroni/Šidák correction and (ii) volume-of-tube formula. We provide extensions of this formula in higher dimensions and show that the width of the bands are expanding with increasing dimensionality. Finally, the results for the regression case will be extended to the classification case. Contributions are made in Section 7.2 and Section 7.3.

### 7.1 Introduction

Nowadays, nonparametric function estimators have become very popular data analytic tools in various fields, see e.g. Tsybakov (2009), Sun et al. (2010) and Ong et al. (2010). Many of their properties have been rigorously investigated and are well understood. An important tool accompanying these estimators is the construction of interval estimates e.g. confidence intervals. In the area of kernel based regression, a popular tool to construct interval estimates is the bootstrap (see e.g. Hall, 1992). This technique produces very accurate intervals at the expense of heavy computational burden.

In the field of neural networks, Chryssolouris et al. (1996) and Papadopoulos et al. (2001) have proposed confidence and prediction intervals. In case of nonlinear regression Rivals and Personnaz (2000) proposed confidence intervals, based on least squares estimation, using the linear Taylor expansion of the nonlinear model. Excellent books discussing confidence intervals for nonlinear regression are written by Seber and Wild (2003) and Ritz and Streibig (2008).

Early works of Bishop and Qazaz (1997) and Goldberg et al. (1998) address the construction of interval estimates via a Bayesian approach and a Markov Chain Monte Carlo method to approximate the posterior noise variance. An extension of the latter was proposed by Kersting et al. (2007). In general, Bayesian intervals (which are often called Bayesian credible intervals) do not exactly coincide with frequentist confidence intervals (as discussed in this Chapter) for two reasons: first, credible intervals incorporate problem-specific contextual information from the prior distribution whereas confidence intervals are only based on the data and second, credible intervals and confidence intervals treat nuisance parameters in radically different ways (see Bernardo and Smith (2000) and references therein).

In this Chapter, we will address some of the difficulties in constructing these interval estimates as well as develop a methodology for interval estimation in case of least squares support vector machines (LS-SVM) for regression which is not based on bootstrap.

Consider the bivariate data  $\{(X_1, Y_1), \dots, (X_n, Y_n)\}$  which form an independent and identically distributed (i.i.d.) sample from a population  $(X, Y)$ . Our interest is to estimate the regression function  $m(X) = \mathbf{E}[Y|X]$  (see Chapter 2), with  $\mathbf{E}[Y|X] = \int y f_{Y|X}(y|x) dy$  where  $f_{Y|X}$  is the conditional distribution of the random variables  $(X, Y)$ . We regard the data as being generated from the model

$$Y = m(X) + \sigma(X)e, \quad (7.1)$$

where  $\mathbf{E}[e|X] = 0$ ,  $\mathbf{Var}[e|X] = \mathbf{E}[e^2|X] - \mathbf{E}^2[e|X] = 1$  and  $X$  and  $e$  are independent. In setting (7.1), it is immediately clear that  $\mathbf{E}[Y|X] = m(X)$  and  $\mathbf{Var}[Y|X] = \sigma^2(X) > 0$ . Two possible situations can occur: (i)  $\sigma^2(X) = \sigma^2 = \text{constant}$  and (ii) the variance is a function of the random variable  $X$ . The first is called homoscedasticity and the latter heteroscedasticity.

One of the problems we will address is the construction of uniform or simultaneous confidence intervals for  $m$ . Specifically, given  $\alpha \in (0, 1)$  and an estimator  $\hat{m}_n$  for  $m$ , we want to find a bound  $g_\alpha$  such that

$$\mathbf{P} \left[ \sup_x |\hat{m}_n(x) - m(x)| \leq g_\alpha \right] \geq 1 - \alpha, \quad (7.2)$$

at least for large sample sizes.

A major difficulty in finding a solution to (7.2) is the fact that nonparametric estimators for  $m$  are biased (kernel estimators in particular). As a consequence,

confidence interval procedures must deal with estimator bias to ensure that the interval is correctly centered and proper coverage is attained (Eubank and Speckman, 1993). In case of an unbiased estimator, we refer to the books of Rice (1995) and Draper and Smith (1998) for the construction of confidence intervals.

In order to avoid the bias estimation problem, several authors have studied the limiting distribution of  $\sup_x |\hat{m}_n(x) - m(x)|$  for various estimators  $\hat{m}_n$  of  $m$ . A pioneering article in this field is due to Bickel and Rosenblatt (1973) for kernel density estimation. Extensions of Bickel and Rosenblatt (1973) to kernel regression are given in Johnston (1982), Hall and Titterington (1988) and Härdle (1989).

A second way to avoid calculating the bias explicitly is to undersmooth. If we smooth less than the optimal amount, then the bias will decrease asymptotically relative to the variance. Hall (1992) showed theoretically that undersmoothing in combination with a pivotal statistic based on bootstrap results into the lowest reduction in coverage error of confidence intervals. Unfortunately, a simple and practical rule for choosing just the right amount of undersmoothing does not seem to exist.

A third and more practical way is to be satisfied with indicating the level of variability involved in a nonparametric regression estimator, without attempting to adjust for the inevitable presence of bias. Bands of this type are easier to construct but require careful interpretation. Formally, the bands indicate pointwise variability intervals for  $\mathbf{E}[\hat{m}_n(X)|X]$ . Based on this idea, often misconceptions exist between confidence intervals and error bars (Wasserman, 2006).

Finally, if it is possible to obtain a reasonable bias estimate it can be used to construct confidence intervals for  $m$ . The application of this approach can be found in local polynomial regression (Fan and Gijbels, 1995, 1996) where a bias estimate can be easily obtained.

Applications of confidence intervals can be found in e.g. the chemical industry, fault detection/diagnosis and system identification/control. These intervals give the user the ability to see how well a certain model explains the true underlying process while taking statistical properties of the estimator into account. In control applications, these intervals are used for robust design while the applicability of these intervals in fault detection are based upon reducing the number of false alarms. For further reading regarding this topic we refer to the books of Isermann (2005) and Witczak (2007).

## 7.2 Estimation of Bias and Variance

### 7.2.1 LS-SVM Regression and Smoother Matrix

By noticing that LS-SVM is a linear smoother (see Definition 3.1), suitable bias and variance formulations can be found. On training data, LS-SVM can be written in matrix form as  $\hat{m}_n = LY$ , where  $\hat{m}_n = (\hat{m}_n(X_1), \dots, \hat{m}_n(X_n))^T$  and  $L$  is a smoother matrix whose  $i^{\text{th}}$  row is  $L(X_i)^T$ , thus  $L_{ij} = l_j(X_i)$ . The entries of the  $i^{\text{th}}$  row show the weights given to each  $Y_i$  in forming the estimate  $\hat{m}_n(X_i)$ . We can state the following theorem.

**Theorem 7.1** *The LS-SVM estimate (2.16) can be written as*

$$\hat{m}_n(x) = \sum_{i=1}^n l_i(x) Y_i \quad (7.3)$$

with  $L(x) = (l_1(x), \dots, l_n(x))^T$  the smoother vector and

$$L(x) = \left[ \Omega_x^{*T} \left( Z^{-1} - Z^{-1} \frac{J_n}{c} Z^{-1} \right) + \frac{1_n^T}{c} Z^{-1} \right]^T, \quad (7.4)$$

with  $\Omega_x^* = (K(x, X_1), \dots, K(x, X_n))^T$  the kernel vector evaluated at point  $x$ ,  $c = 1_n^T \left( \Omega + \frac{I_n}{\gamma} \right)^{-1} 1_n$ ,  $Z = \Omega + \frac{I_n}{\gamma}$ ,  $J_n$  a square matrix with all elements equal to 1 and  $1_n = (1, \dots, 1)^T$ .

Then the estimator, under model (2.16), has conditional mean

$$\mathbf{E}[\hat{m}_n(x)|X = x] = \sum_{i=1}^n l_i(x) m(x_i)$$

and conditional variance

$$\mathbf{Var}[\hat{m}_n(x)|X = x] = \sum_{i=1}^n l_i(x)^2 \sigma^2(x_i). \quad (7.5)$$

PROOF. In matrix form, the resulting LS-SVM model (2.16) on training data is given by

$$\hat{m}_n = \Omega \hat{\alpha} + 1_n \hat{b}. \quad (7.6)$$

Solving the linear system (2.15) yields

$$\hat{\alpha} = \left( \Omega + \frac{I_n}{\gamma} \right)^{-1} (Y - 1_n \hat{b})$$

and

$$\hat{b} = \frac{1_n^T \left( \Omega + \frac{I_n}{\gamma} \right)^{-1} Y}{1_n^T \left( \Omega + \frac{I_n}{\gamma} \right)^{-1} 1_n}.$$

Plugging the expressions for  $\hat{a}$  and  $\hat{b}$  into (7.6) gives

$$\begin{aligned} \hat{m}_n &= \left[ \Omega \left( Z^{-1} - Z^{-1} \frac{J_n}{c} Z^{-1} \right) + \frac{J_n}{c} Z^{-1} \right] Y \\ &= LY, \end{aligned}$$

with  $c = 1_n^T \left( \Omega + \frac{I_n}{\gamma} \right)^{-1} 1_n$ ,  $Z = \Omega + \frac{I_n}{\gamma}$ ,  $J_n$  is a square matrix with all elements equal to 1. The above derivation is valid when all points  $x$  are considered as training data. However, evaluating LS-SVM in an arbitrary point  $x$  can be written as follows

$$\begin{aligned} \hat{m}_n(x) &= \Omega_x^{*T} \hat{a} + 1_n \hat{b} \\ &= \left[ \Omega_x^{*T} \left( Z^{-1} - Z^{-1} \frac{J_n}{c} Z^{-1} \right) + \frac{1_n^T}{c} Z^{-1} \right] Y \\ &= L(x)^T Y, \end{aligned}$$

with  $\Omega_x^* = (K(x, X_1), \dots, K(x, X_n))^T$  the kernel vector evaluated in an arbitrary point  $x$  and  $1_n = (1, \dots, 1)^T$ .

The conditional mean and conditional variance of the LS-SVM can then be derived as follows

$$\begin{aligned} \mathbf{E}[\hat{m}_n(x)|X = x] &= \sum_{i=1}^n l_i(x) \mathbf{E}[Y_i|X = x_i] \\ &= \sum_{i=1}^n l_i(x) m(x_i) \end{aligned}$$

and

$$\begin{aligned} \mathbf{Var}[\hat{m}_n(x)|X = x] &= \sum_{i=1}^n l_i(x)^2 \mathbf{Var}[Y_i|X = x_i] \\ &= \sum_{i=1}^n l_i(x)^2 \sigma^2(x_i). \end{aligned}$$

□

**Remark** Note that one should not confuse linear smoothers i.e. smoothers of the form (7.3) such as NW, local polynomial regression, splines, LS-SVM, wavelets etc. with linear regression, in which one assumes that the regression function  $m$  is linear.

## 7.2.2 Bias Estimation

Using Theorem 7.1, the conditional bias can be written as

$$b(x) = \mathbf{E}[\hat{m}_n(x)|X = x] - m(x) = \sum_{i=1}^n l_i(x)m(x_i) - m(x).$$

It can be shown that, in the one dimensional case, by using a Taylor series expansion around the fitting point  $x \in \mathbb{R}$ , that for  $|x_i - x| \leq h$  the conditional bias is equal to

$$\mathbf{E}[\hat{m}_n(x)|X = x] - m(x) = m'(x) \sum_{i=1}^n (x_i - x)l_i(x) + \frac{m''(x)}{2} \sum_{i=1}^n (x_i - x)^2 l_i(x) + o(\vartheta(h, \gamma)),$$

where  $\vartheta$  is an unknown function describing the relation between the two tuning parameters.

Although the above expression gives insight on how the conditional bias behaves asymptotically, it is quite hard to use this in practice since it involves the estimation of first and second order derivatives of the unknown  $m$ . In fact, this procedure can be rather complicated in the multivariate case, especially when estimating derivatives.

Therefore, we opt for a procedure which does not rely completely on the asymptotic expression, but stays “closer” to the exact expression for the conditional bias. As a result, this will carry more information about the finite sample bias.

**Theorem 7.2** Let  $L(x)$  be the smoother vector evaluated in a point  $x$  and denote  $\hat{m}_n = (\hat{m}_n(X_1), \dots, \hat{m}_n(X_n))^T$ . Then, the estimated conditional bias for LS-SVM is given by

$$\hat{b}(x) = \widehat{\text{bias}}[\hat{m}_n(x)|X = x] = L(x)^T \hat{m}_n - \hat{m}_n(x). \quad (7.7)$$

PROOF. The exact conditional bias for LS-SVM is given by (in matrix form)

$$\mathbf{E}[\hat{m}_n|X] - m = (L - I_n)m,$$



where  $m = (m(X_1), \dots, m(X_n))^T$  and  $\hat{m}_n = LY$ . Observe that the residuals are given by

$$\hat{\epsilon} = Y - \hat{m}_n = (I_n - L)Y.$$

Taking expectations yields

$$\begin{aligned} \mathbf{E}[\hat{\epsilon}|X] &= m - Lm \\ &= -\text{bias}[\hat{m}_n|X]. \end{aligned}$$

This suggests estimating the conditional bias by smoothing the negative residuals

$$\begin{aligned} \widehat{\text{bias}}[\hat{m}_n|X] &= -L\hat{\epsilon} \\ &= -L(I_n - L)Y \\ &= (L - I_n)\hat{m}_n. \end{aligned}$$

Therefore, evaluating the estimated conditional bias at a point  $x$  can be written as

$$\begin{aligned} \hat{b}(x) = \widehat{\text{bias}}[\hat{m}_n(x)|X = x] &= \sum_{i=1}^n l_i(x)\hat{m}_n(x_i) - \hat{m}_n(x) \\ &= L(x)^T \hat{m}_n - \hat{m}_n(x). \end{aligned}$$

□

Techniques as (7.7) are known as plug-in bias estimates and can be directly calculated from the LS-SVM estimate (see also Cornillon et al., 2009). However, it is possible to construct better bias estimates, at the expense of extra calculations, by using a technique called double smoothing (Härdle et al., 1992) which can be seen as a generalization of the plug-in based technique. Before explaining the double smoothing, we need to introduce the following definition.

**Definition 7.1 (Jones and Foster, 1993)** *A kernel  $K$  is called a  $k^{\text{th}}$ -order kernel if*

$$\begin{cases} \int K(u) du = 1 \\ \int u^j K(u) du = 0, \quad j = 1, \dots, k-1 \\ \int u^k K(u) du \neq 0. \end{cases}$$

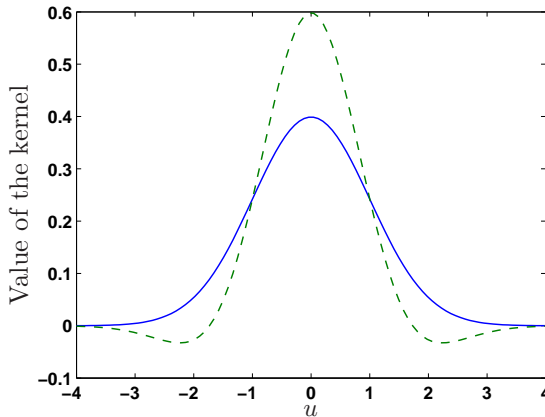
For example, the Gaussian kernel satisfies this condition for  $k = 2$  but the linear and polynomial kernels do not. There are several rules for constructing higher-order kernels, see e.g. Jones and Foster (1993). Let  $K_{[k]}$  be a  $k^{\text{th}}$ -order symmetric kernel ( $k$  even) which is assumed to be differentiable. Then

$$K_{[k+2]}(u) = \frac{k+1}{k} K_{[k]}(u) + \frac{1}{k} u K'_{[k]}(u), \quad (7.8)$$

is a  $(k + 2)^{\text{th}}$ -order kernel. Hence, this formula can be used to generate higher-order kernels in an easy way. Consider for example the standard normal density function  $\phi$  which is a second-order kernel. Then a fourth-order kernel can be obtained via (7.8):

$$\begin{aligned} K_{[4]}(u) &= \frac{3}{2}\phi(u) + \frac{1}{2}u\phi'(u) \\ &= \frac{1}{2}(3 - u^2)\phi(u) \\ &= \frac{1}{2} \frac{1}{\sqrt{2\pi}} (3 - u^2) \exp\left(-\frac{u^2}{2}\right). \end{aligned} \quad (7.9)$$

In the remaining of this Chapter, the Gaussian kernel (second and fourth order) will be used. Figure 7.1 shows the standard normal density function  $\phi$  together with the fourth-order kernel  $K_{[4]}$  derived from it. It can be easily verified using Bochner's lemma (Bochner, 1959) that  $K_{[4]}$  is an admissible positive definite kernel.



**Figure 7.1:** Plot of  $K_{[2]}(u) = \phi(u)$  (solid curve) and  $K_{[4]}(u)$  (dashed curve).

The idea of double smoothing bias estimation is then given by Müller (1985) and Härdle et al. (1992).

**Proposition 7.1 (Double Smoothing)** *Let  $L(x)$  be the smoother vector evaluated in a point  $x$ , let  $\hat{m}_{n,g} = (\hat{m}_{n,g}(X_1), \dots, \hat{m}_{n,g}(X_n))^T$  be another LS-SVM smoother based on the same data set and a  $k^{\text{th}}$ -order kernel with  $k > 2$  and different bandwidth  $g$ . Then, the double smoothing bias estimate of the conditional bias for*

*LS-SVM is defined by*

$$\hat{b}(x) = L(x)^T \hat{m}_{n,g} - \hat{m}_{n,g}(x). \quad (7.10)$$

A criticism on this approach is that it requires selection of another bandwidth  $g$ . Härdle et al. (1992) suggested to take  $g$  of larger order than  $n^{-1/9}$  but of smaller order than  $n^{-1/8}$  in case of a combination of a 2<sup>th</sup> and 4<sup>th</sup>-order kernel. Recently, Beran et al. (2009) proposed an iterative data-driven approach to find the bandwidth  $g$ .

Note that the bias estimate (7.10) can be thought of as an iterated smoothing algorithm. The pilot smooth  $\hat{m}_{n,g}$  (with fourth-order kernel  $K_{[4]}$  and bandwidth  $g$ ) is resmoothed with kernel  $K$  (Gaussian kernel), incorporated in the smoother matrix  $L$ , and bandwidth  $h$ . Because smoothing is done twice, it is called double smoothing. A generalization of double smoothing, called iterative smoothing, is proposed by Cornillon et al. (2009).

### 7.2.3 Variance Estimation

Our goal is to derive a fully-automated procedure to estimate the variance function  $\sigma^2$ . Due to the simple decomposition  $\sigma^2(x) = \mathbf{Var}[Y|X = x] = \mathbf{E}[Y^2|X = x] - \{\mathbf{E}[Y|X = x]\}^2$ , one tends to use the following obvious and direct estimator (Härdle and Tsybakov, 1997)

$$\hat{\sigma}_d^2(x) = \mathbf{E}[Y^2|X = x] - \{\hat{m}_n(x)\}^2.$$

However, there are some drawbacks in using such an estimator. For example,  $\hat{\sigma}_d^2(x)$  is not always non-negative due to estimation error, especially if different smoothing parameters are used in estimating the regression function and  $\mathbf{E}[Y^2|X]$ . Furthermore, such a method can result into a very large bias (see Fan and Yao, 1998). Before stating our proposed estimator, we first need a condition on the weights of the LS-SVM (see Lemma 7.1). The resulting variance estimator is given in (7.12) by making use of Theorem 7.3.

**Lemma 7.1** *The weights  $\{l_i(x)\}$  of the LS-SVM smoother in an arbitrary point  $x$  are normal i.e.*

$$\sum_{i=1}^n l_i(x) = 1.$$

PROOF. Lemma 7.1 will be proven if we show that  $L1_n = 1_n$ . Using the expression for the smoother matrix (7.4) for LS-SVM

$$\begin{aligned} L1_n &= \left[ \Omega \left( Z^{-1} - Z^{-1} \frac{J_n}{c} Z^{-1} \right) + \frac{J_n}{c} Z^{-1} \right] 1_n \\ &= \Omega \left( Z^{-1} - Z^{-1} \frac{J_n}{c} Z^{-1} \right) 1_n + \frac{J_n}{c} Z^{-1} 1_n \\ &= \Omega Z^{-1} \left( 1_n - \frac{J_n}{c} Z^{-1} 1_n \right) + \frac{J_n}{c} Z^{-1} 1_n. \end{aligned}$$

It suffices to show that  $\frac{J_n}{c} Z^{-1} 1_n = 1_n$  to complete the proof.

$$\frac{J_n}{c} Z^{-1} 1_n = \frac{1_n 1_n^T \left( \Omega + \frac{I_n}{\gamma} \right)^{-1} 1_n}{1_n^T \left( \Omega + \frac{I_n}{\gamma} \right)^{-1} 1_n} = 1_n.$$

We can now formulate the result for any arbitrary point  $x$ . Let  $L(x)$  be the smoother vector in an arbitrary point  $x$ , then

$$\begin{aligned} \sum_{i=1}^n l_i(x) &= L(x)^T 1_n \\ &= \left[ \Omega_x^{*T} \left( Z^{-1} - Z^{-1} \frac{J_n}{c} Z^{-1} \right) + \frac{1_n^T}{c} Z^{-1} \right] 1_n. \end{aligned}$$

Similar to the derivation given above, we have to show that  $\frac{1_n^T}{c} Z^{-1} 1_n = 1$  to conclude the proof.

$$\frac{1_n^T}{c} Z^{-1} 1_n = \frac{1_n^T \left( \Omega + \frac{I_n}{\gamma} \right)^{-1} 1_n}{1_n^T \left( \Omega + \frac{I_n}{\gamma} \right)^{-1} 1_n} = 1.$$

□

**Theorem 7.3 (Variance Estimation)** Assume model (7.1), let  $L$  denote the smoother matrix corresponding to the initial smooth. Let  $S$  denote the smoother matrix corresponding to a natural way of estimating  $\sigma^2(\cdot)$  based on smoothing the squared residuals. Denote by  $S(x)$  the smoother vector in an arbitrary point  $x$ .

Assume that  $S$  preserves constant vectors i.e.  $S\mathbf{1}_n = \mathbf{1}_n$ , then an estimator for the variance function  $\sigma^2(\cdot)$ , evaluated in an arbitrary point  $x$ , is given by

$$\hat{\sigma}^2(x) = \frac{S(x)^T \text{diag}(\hat{e}\hat{e}^T)}{1 + S(x)^T \text{diag}(LL^T - L - L^T)}, \quad (7.11)$$

where  $\hat{e}$  denote the residuals and  $\text{diag}(A)$  is the column vector containing the diagonal entries of the square matrix  $A$ .

PROOF. Let  $L$  be the smoother matrix corresponding to an initial smooth of the data and set

$$\hat{e} = (I_n - L)Y$$

the vector of residuals. Then, a natural way of estimating the variance function  $\sigma^2(\cdot)$  is to smooth the squared residuals to obtain  $S \text{diag}(\hat{e}\hat{e}^T)$ . It is also reasonable that the estimator should be unbiased when the errors are homoscedastic. Thus, under homoscedasticity, set  $\Sigma' = \mathbf{E}[(Y - m)^2|X]$  and  $B_1 = \mathbf{E}[LY|X] - m$ , we obtain

$$\begin{aligned} \mathbf{E}[S \text{diag}(\hat{e}\hat{e}^T)|X] &= S \mathbf{E}[\text{diag}\{(I_n - L)YY^T(I_n - L)^T\}|X] \\ &= S [\text{diag}\{(I_n - L)\mathbf{E}(YY^T|X)(I_n - L)^T\}] \\ &= S [\text{diag}\{(I_n - L)(mm^T + \Sigma')(I_n - L)^T\}] \\ &= S [\text{diag}\{(I_n - L)mm^T(I_n - L)^T + (I_n - L)\Sigma'(I_n - L)^T\}] \\ &= S [\text{diag}(B_1B_1^T) + \sigma^2 \text{diag}((I_n - L)(I_n - L)^T)] \\ &= S [\text{diag}(B_1B_1^T) + \sigma^2(1_n + \Delta)], \end{aligned}$$

where  $\Delta = \text{diag}(LL^T - L - L^T)$ . When  $\hat{m}_n$  is conditionally unbiased, i.e.  $B_1 = 0$ ,  $\mathbf{E}[S \text{diag}(\hat{e}\hat{e}^T)|X] = \sigma^2(1_n + S\Delta)$ . Using Lemma 7.1, motivates the following variance estimator at an arbitrary point  $x$

$$\hat{\sigma}^2(x) = \frac{S(x)^T \text{diag}(\hat{e}\hat{e}^T)}{1 + S(x)^T \text{diag}(LL^T - L - L^T)}, \quad (7.12)$$

where  $S(x)$  is the smoother matrix, based on smoothing the squared residuals, in an arbitrary point  $x$ . □

For other nonparametric estimates of  $\hat{\sigma}^2(x)$  see e.g. Müller and Stadtmüller (1987), Neumann (1994), Stadtmüller and Tsybakov (1995), Müller et al. (2003) and Kohler (2006).

The class of variance function estimators (7.11) can be viewed as a generalization of those commonly used in parametric modeling (see e.g. Rice, 1995). This can easily

be seen as follows. Consider a linear model in the heteroscedastic case, then one should replace the smoother matrix  $L$  by the hat matrix  $Q = X'(X'^T X')^{-1} X'^T$  where  $X'$  denotes the  $n \times p$  design matrix with  $p$  the number of parameters to be estimated in the model. Since  $Q$  is symmetric and idempotent, (7.11) reduces to

$$\hat{\sigma}^2(x) = \frac{S(x)^T \text{diag}(\hat{e}\hat{e}^T)}{1 - S(x)^T \text{diag}(Q)}.$$

On the other hand, considering a homoscedastic nonparametric regression model (and by taking a smoother resulting in a symmetric smoother matrix  $L$ ), then averaging the squared residuals, by taking  $S = n^{-1}1_n 1_n^T$ , will result in the following estimator of the error variance

$$\hat{\sigma}^2 = \frac{Y^T(L - I_n)^T(L - I_n)Y}{n + \text{tr}(LL^T - 2L)},$$

which includes variance estimators for nonparametric regression considered by e.g. Buckley et al. (1988). For the homoscedastic linear regression model, the estimator reduces to well-known estimator of the error variance

$$\hat{\sigma}^2 = \frac{Y^T(I_n - Q)Y}{n - p}.$$

All the above estimators of error variance are model based, i.e. either based on a parametric or nonparametric model. For the homoscedastic case, model-free error variance estimators are proposed by Rice (1984), Gasser et al. (1986), Hall and Marron (1990) and Pelckmans et al. (2005).

Next we approximate the conditional variance of LS-SVM (7.5). Given the estimator for the error variance function (7.11), then an estimate of the conditional variance of LS-SVM with heteroscedastic errors is given by

$$\hat{V}(x) = \widehat{\mathbf{Var}}[\hat{m}_n(x)|X = x] = L(x)^T \hat{\Sigma}^2 L(x), \quad (7.13)$$

with  $\hat{\Sigma}^2 = \text{diag}(\hat{\sigma}^2(X_1), \dots, \hat{\sigma}^2(X_n))$ .

## 7.3 Confidence and Prediction Intervals: Regression

### 7.3.1 Pointwise Confidence Intervals

The estimated bias (7.10) and variance (7.13) can be used to construct pointwise Confidence Intervals (CIs). In the next theorem we show the asymptotic normality of LS-SVM by using a central limit theorem for sums of independent random variables, see Gnedenko and Kolmogorov (1968), Serfling (1980), Shiryaev (1996) and Petrov (2004). The following derivation is not only valid for LS-SVM, but for all linear smoothers.

**Theorem 7.4 (Asymptotic Normality)** *Let  $\xi_1, \dots, \xi_n$ , with  $\xi_k = l_k(x)Y_k$ , be a sequence of independent random variables with finite second moments. Let  $\mathbf{Var}[\xi_k|X = x_k] > 0$  and  $\hat{m}_n(x) = \xi_1 + \dots + \xi_n$ . Let the error variance function  $\sigma^2(x_k) > 0$  for all  $k$ . Denote the conditional bias and variance of  $\hat{m}_n(x)$  by  $b(x)$  and  $V(x)$  respectively. Further, assume that  $\mathbf{P}[|Y| \leq M] = 1$  with  $M < \infty$ . Then, for*

$$\max_{1 \leq k \leq n} \frac{|l_k(x)|}{\|l(x)\|} \rightarrow 0, \quad n \rightarrow \infty,$$

it follows that

$$\frac{\hat{m}_n(x) - m(x) - b(x)}{\sqrt{V(x)}} \xrightarrow{d} \mathcal{N}(0,1).$$

PROOF. First, let  $\mu_k = \mathbf{E}[\xi_k|X = x_k]$  and set  $D_n^2 = \sum_{k=1}^n \mathbf{Var}[\xi_k|X = x_k] = \sum_{k=1}^n l_k^2(x)\sigma^2(x_k)$ . Since we have a triangular array of random variables, the theorem will be proven if the Lindeberg condition is satisfied: for every  $\varepsilon > 0$

$$\frac{1}{D_n^2} \sum_{k=1}^n \int_{\{z:|z-\mu_k| \geq \varepsilon D_n\}} (z - \mu_k)^2 dF_k(z) \rightarrow 0, \quad n \rightarrow \infty.$$

First, notice that for any  $k$  the following two statements are valid

$$|\xi_k| = |l_k(x)Y_k| = |l_k(x)||Y| \leq M|l_k(x)|$$

and therefore

$$|\xi_k - \mu_k| \leq |\xi_k| + |\mu_k| \leq |\xi_k| + \mathbf{E}[|\xi_k|X = x] \leq 2M|l_k(x)|.$$

Second, by Chebyshev's inequality

$$\begin{aligned} \int_{\{z:|z-\mu_k| \geq \varepsilon D_n\}} (z - \mu_k)^2 dF_k(z) &= \mathbf{E} [(\xi_k - \mu_k)^2 I_{(|\xi_k - \mu_k| \geq \varepsilon D_n)}] \\ &\leq 4M^2|l_k(x)|^2 \mathbf{P}[ (|\xi_k - \mu_k| \geq \varepsilon D_n) ] \\ &\leq 4M^2|l_k(x)|^2 \frac{\sigma^2(x_k)l_k^2(x)}{\varepsilon^2 D_n^2}, \end{aligned}$$

Consequently,

$$\begin{aligned}
 \frac{1}{D_n^2} \sum_{k=1}^n \int_{\{z: |z-\mu_k| \geq \varepsilon D_n\}} (z - \mu_k)^2 dF_k(z) &\leq \frac{1}{D_n^2} \sum_{k=1}^n 4M^2 |l_k(x)|^2 \frac{\sigma^2(x_k) l_k^2(x)}{\varepsilon^2 D_n^2} \\
 &\leq \frac{4M^2}{\varepsilon^2 D_n^4} \max_{1 \leq k \leq n} |l_k(x)|^2 \sum_{k=1}^n \sigma^2(x_k) l_k^2(x) \\
 &\leq \frac{4M^2}{\varepsilon^2 \min_{1 \leq k \leq n} \sigma^2(x_k)} \max_{1 \leq k \leq n} \frac{|l_k(x)|^2}{\|l(x)\|^2}
 \end{aligned}$$

Hence, the Lindeberg condition

$$\frac{1}{D_n^2} \sum_{k=1}^n \int_{\{z: |z-\mu_k| \geq \varepsilon D_n\}} (z - \mu_k)^2 dF_k(z) \rightarrow 0, \quad n \rightarrow \infty,$$

is satisfied and therefore, the theorem is verified.  $\square$

Conditions of the form

$$\max_{1 \leq k \leq n} \frac{|l_k(x)|}{\|l(x)\|}, \quad (7.14)$$

have been studied in Shao (2003) and Bhansali et al. (2006). They show that condition (7.14) goes to zero, for  $n \rightarrow \infty$ , if the largest eigenvalue of the smoother matrix  $L$  is dominated by all other eigenvalues together. In case of linear smoothers with a simpler smoother matrix, e.g. Priestley-Chao and NW, condition 7.14 is satisfied for  $nh^d \rightarrow \infty$  as  $n \rightarrow \infty$  and  $f(x) > 0$ . Figure 7.2 illustrates condition (7.14) as a function of the number of data points for LS-SVM. This empirically confirms that when  $n$  is large, condition (7.14) will become small.

With the estimated bias and variance, an approximate  $100(1 - \alpha)\%$  pointwise CI for  $\hat{m}_n(x)$  is

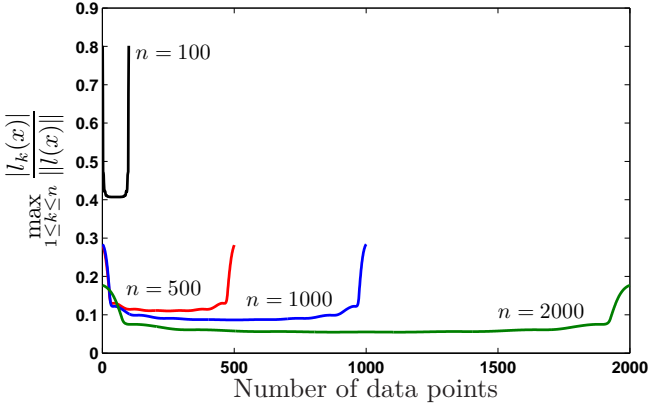
$$\hat{m}_n(x) - \hat{b}(x) \pm z_{1-\alpha/2} \sqrt{\hat{V}(x)}, \quad (7.15)$$

where  $z_{1-\alpha/2}$  denotes the  $(1-\alpha/2)^{\text{th}}$  quantile of the standard Gaussian distribution. This approximate CI is valid if

$$\frac{\hat{V}(x)}{V(x)} \xrightarrow{P} 1 \quad \text{and} \quad \frac{\hat{b}(x)}{b(x)} \xrightarrow{P} 1.$$

This in turn requires a different bandwidth used in assessing the bias and variance (Fan and Gijbels, 1996), as we have done in Section 7.2.





**Figure 7.2:** Condition (7.14) as a function of the number of data points  $n$  for LS-SVM.

The previous approach can also be easily applied for FS-LSSVM (see Chapter 4). By replacing

$$L = \hat{\Phi}_e \left( \hat{\Phi}_e^T \hat{\Phi}_e + \frac{I_{m+1}}{\gamma} \right)^{-1} \hat{\Phi}_e^T,$$

i.e. the smoother matrix of FS-LSSVM, all the above derivations remain valid.

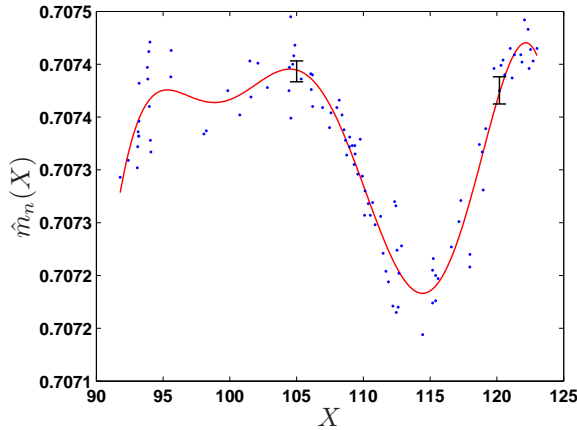
### 7.3.2 Simultaneous Confidence Intervals

The confidence intervals presented so far in this Section are pointwise. However, in many cases one is often more interested in simultaneous or uniform CIs. With two simple examples we can state the difference between both types of CIs.

**Example 7.1** *By looking at two pointwise confidence intervals, constructed by (7.15) for the Fossil data set (Ruppert et al., 2003), in Figure 7.3 we can make the following two statements **separately***

- (0.70743,0.70745) is an approximate 95% pointwise confidence interval for  $m(105)$ ;
- (0.70741,0.70744) is an approximate 95% pointwise confidence interval for  $m(120)$ .

*However, as is well known in multiple comparison theory, it is wrong to state that  $m(105)$  is contained in (0.70743,0.70745) and simultaneously  $m(120)$  is contained in (0.70741,0.70744) with 95% confidence.*



**Figure 7.3:** Fossil data with two pointwise 95% confidence intervals.

**Example 7.2** Suppose our aim is to estimate some function  $m$ . For example,  $m$  might be the proportion of people of a particular age who support a given candidate in an election. If the data is measured at the precision of a single year, we can construct a “pointwise” 95% confidence interval for each age. Each of these confidence intervals covers the corresponding true value  $m(x)$  with a coverage probability of 0.95. The “simultaneous” coverage probability of a collection of confidence intervals is the probability that all of them cover their corresponding true values simultaneously.

From these two examples, it is clear that it is not correct to connect the pointwise confidence intervals to produce a band or interval around the estimated function. Also, simultaneous confidence bands will be wider than pointwise confidence bands. In order to make these statements simultaneously we have to modify the interval to obtain simultaneous confidence intervals. Mathematically speaking, we are searching for the width of the bands  $c$ , given a confidence level  $\alpha \in (0,1)$ , such that

$$\inf_{m \in \mathcal{F}_{n,p}} \mathbf{P} \left\{ \hat{m}_n(x) - c\sqrt{\hat{V}(x)} \leq m(x) \leq \hat{m}_n(x) + c\sqrt{\hat{V}(x)}, \forall x \in \mathcal{X} \right\} = 1 - \alpha,$$

for some suitable class of smooth functions  $\mathcal{F}_{n,p}$  (see Chapter 2) with  $\hat{m}_n$  an estimate of the true function  $m$  and  $\mathcal{X} \subseteq \mathbb{R}^d$ .

Three major groups exist to determine a suitable width of the bands: (i) Monte Carlo simulations, (ii) Bonferroni/Šidák corrections (Šidák, 1967) or other types, e.g. the length heuristic (Efron, 1997) and (iii) volume-of-tube formula (Weyl, 1939; Rice, 1939; Knafel et al., 1985; Sun and Loader, 1994; Loader, 1999). The

latter was originally formulated as a geometric problem of primary importance by Hotelling at the Mathematics Club at Princeton in the late 1930s.

Although Monte Carlo based modifications are accurate (even when the number of data points  $n$  is relatively small), they are computationally expensive. Therefore, we will not discuss this type of methods in this Chapter. Interested readers can browse through Ruppert et al. (2003) and reference therein.

Šidák and Bonferroni corrections are one of the most popular since they are easy to calculate and produce quite acceptable results. In what follows, the rationale behind the Šidák correction (generalization of Bonferroni) is elucidated. This correction is derived by assuming that individual tests are independent. Let the significance threshold for each test be  $\beta$  (significance level of pointwise confidence interval), then the probability that at least one of the tests is significant under this threshold is  $(1 - \text{the probability that none of them are significant})$ . Since we are assuming that they are independent, the probability that all of them are not significant is the product of the probabilities that each of them is not significant, or  $1 - (1 - \beta)^n$ . Now we want this probability to be equal to  $\alpha$ , the significance level for the entire series of tests (or simultaneous confidence interval). By solving for  $\beta$ , we get  $\beta = 1 - (1 - \alpha)^{1/n}$ . To obtain an approximate  $100(1 - \alpha)\%$  simultaneous confidence intervals, based on a Šidák correction, (7.15) has to be modified into

$$\hat{m}_n(x) - \hat{b}(x) \pm z_{1-\beta/2} \sqrt{\hat{V}(x)}$$

The last method analytically approximates the modification of the interval, i.e. the width  $c$  of the interval or bands. Sun (1993) studied the tail probabilities of suprema of Gaussian random processes. This turns out to be very useful in constructing simultaneous confidence bands. Proposition (7.2) and Proposition (7.3) summarize the results of Sun (1993) and Sun and Loader (1994) in the two and  $d$  dimensional case respectively when the error variance  $\sigma^2$  is not known a priori and has to be estimated. It is important to note that the justification for the tube formula assumes the errors have a normal distribution (this can be relaxed to spherically symmetric distributions, see Loader and Sun (1997)), but does not require letting  $n \rightarrow \infty$ . As a consequence, the tube formula does not require embedding finite sample problems in a possibly artificial sequence of problems, and the formula can be expected to work well at small sample sizes.

**Proposition 7.2 (two-dimensional)** *Suppose  $\mathcal{X}$  is a rectangle in  $\mathbb{R}^2$ . Let  $\kappa_0$  be the area of the continuous manifold  $\mathfrak{M} = \{T(x) = L(x)/\|L(x)\|_2 : x \in \mathcal{X}\}$ , let  $\zeta_0$  be the volume of the boundary of  $\mathfrak{M}$  denoted by  $\partial\mathfrak{M}$ . Let  $T_j(x) = \partial T(x)/\partial x_j, j =$*

$1, \dots, d$  and let  $c$  be the width of the bands. Then,

$$\alpha = \frac{\kappa_0}{\sqrt{\pi^3}} \frac{\Gamma\left(\frac{(\nu+1)}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)} \frac{c}{\sqrt{\nu}} \left(1 + \frac{c^2}{\nu}\right)^{-\frac{\nu+1}{2}} + \frac{\zeta_0}{2\pi} \left(1 + \frac{c^2}{\nu}\right)^{-\frac{\nu}{2}} + \mathbf{P}[|t_\nu| > c] \quad (7.16)$$

$$+ o\left(\exp\left(-\frac{c^2}{2}\right)\right),$$

with  $\kappa_0 = \int_{\mathcal{X}} \det^{1/2}(A^T A) dx$ ,  $\zeta_0 = \int_{\partial\mathcal{X}} \det^{1/2}(A_\star^T A_\star) dx$  where  $A = (T_1(x), \dots, T_d(x))$  and  $A_\star = (T_1(x), \dots, T_{d-1}(x))$ .  $t_\nu$  is a  $t$ -distributed random variable with  $\nu = n - \text{tr}(L)$ , smoother matrix  $L \in \mathbb{R}^{n \times n}$ , degrees of freedom.

**Proposition 7.3 ( $d$ -dimensional)** Suppose  $\mathcal{X}$  is a rectangle in  $\mathbb{R}^d$  and let  $c$  be the width of the bands. Then,

$$\alpha = \kappa_0 \frac{\Gamma\left(\frac{(\nu+1)}{2}\right)}{\pi^{\frac{d+1}{2}}} \mathbf{P}\left[F_{d+1,\nu} > \frac{c^2}{d+1}\right] + \frac{\zeta_0}{2} \frac{\Gamma\left(\frac{d}{2}\right)}{\pi^{\frac{d}{2}}} \mathbf{P}\left[F_{d,\nu} > \frac{c^2}{d}\right] \quad (7.17)$$

$$+ \frac{\kappa_2 + \zeta_1 + z_0}{2\pi} \frac{\Gamma\left(\frac{(\nu-1)}{2}\right)}{\pi^{\frac{d-1}{2}}} \mathbf{P}\left[F_{d-1,\nu} > \frac{c^2}{d-1}\right]$$

$$+ O\left(c^{d-4} \exp\left(-\frac{c^2}{2}\right)\right),$$

where  $\kappa_2$ ,  $\zeta_1$  and  $z_0$  are certain geometric constants.  $F_{q,\nu}$  is an  $F$ -distributed random variable with  $q$  and  $\nu = \eta_1^2/\eta_2$  degrees of freedom (Cleveland and Devlin, 1988) where  $\eta_1 = \text{tr}[(I_n - L^T)(I_n - L)]$  and  $\eta_2 = \text{tr}[(I_n - L^T)(I_n - L)]^2$ .

Equations (7.16) and (7.17) contain quantities which are often rather difficult to compute in practice. Therefore, the following approximations can be made: (i) according to Loader (1999),  $\kappa_0 = \frac{\pi}{2}(\text{tr}(L) - 1)$  and (ii) it is shown in the simulations of Sun and Loader (1994) that the third term is negligible in (7.17). In this thesis, we neglected the third term and calculated the other quantities by means of quasi Monte Carlo integration based on lattice rules (Nuyens, 2007). More details on the computation of these constants can be found in Sun and Loader (1994) and Ruppert et al. (2003). Finally, to compute the value  $c$ , the width of the bands, any method for solving nonlinear equations can be used.

**Remark** Fortunately, in the one dimensional case the volume-of-formula can be greatly simplified. The width of the bands is approximately given by (Ruppert et al., 2003; Wasserman, 2006)

$$c = \sqrt{2 \log\left(\frac{\kappa_0}{\alpha\pi}\right)} \quad (7.18)$$

where

$$\kappa_0 = \int \frac{\sqrt{\|L(x)\|^2 \|L'(x)\|^2 - (L(x)^T L'(x))^2}}{\|L(x)\|^2} dx,$$

where  $L'(x) = (d/dx)L(x)$ , with the differentiation applied elementwise.

**Remark** We would like to emphasize that the volume-of-tube formula was derived for the homoscedastic case. In the linear regression context, some modifications of the volume-of-tube formula have been developed to adjust for cases with heteroscedastic errors (Faraway and Sun, 1995) and correlated errors (Sun et al., 1999), whereas (Sun et al., 2000) considered generalized linear models. Extensions to handling of correlated and heteroscedastic data in a nonparametric framework is still an open problem.

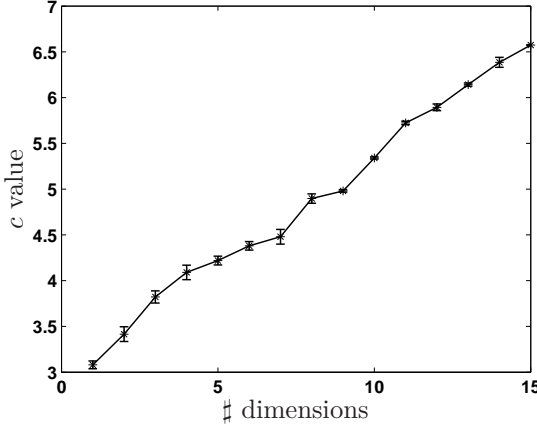
To illustrate the effect of increasing dimensionality on the  $c$  value in (7.17), we conduct the following Monte Carlo study. For increasing dimensionality, we calculate the  $c$  value for a Gaussian density function. 1000 data points were generated uniformly on  $[-3,3]^d$ . Figure 7.4 shows the result of the calculated  $c$  value averaged over 20 times for each dimension. It can be seen that the width of the bands is increasing for increasing dimensionality. Theoretical derivations confirming this simulation can be found in (Hastie et al., 2009; Vapnik, 1999). Simply put, estimating a regression function in a high-dimensional space is especially difficult because it is not possible to densely pack the space with finitely many sample points (Györfi et al., 2002), see also the curse of dimensionality in Chapter 2. The uncertainty of the estimate is becoming larger for increasing dimensionality, hence the confidence bands are wider.

For the Fossil data set ( $n = 106$ ), set  $\alpha = 0.05$ , then  $z_{1-\alpha/2} = 1.96$  and  $z_{1-\beta/2} = 3.49$ . The simultaneous intervals, obtained by using a Šidák correction, are about 1.78 ( $= 3.49/1.96$ ) times wider than the pointwise intervals. Monte Carlo simulations and the volume-of-tube formula (7.18) resulted in a width  $c$  of 3.2 and 3.13 respectively. This is the reason why Šidák (and also Bonferroni) corrections are often said to produce conservative confidence intervals.

Finally,  $100(1 - \alpha)\%$  bias-corrected simultaneous CI are of the form

$$\hat{m}_n(x) - \hat{b}(x) \pm c\sqrt{\hat{V}(x)}. \tag{7.19}$$

Unfortunately, Sun and Loader (1994) and Loader (1999) showed that CIs of form (7.19) cannot work well. Suppose  $\hat{m}_n$  is a reasonably efficient estimate of  $m$ . Then, subtracting the bias estimate  $\hat{b}$  will generally increase variance more than it reduces bias, even if the bias is estimated by the principle of double smoothing. Sun and Loader (1994) reported that this correction fails badly, sometimes being worse than no correction (low coverage probability). Similar results were obtained by Härdle and Marron (1991). Therefore, we use the proposition of Sun and Loader (1994) to attain the required coverage probability. Let  $\mathcal{F}_{n,p,\delta}$  be the class



**Figure 7.4:** Result of the calculated  $c$  value averaged over 20 times for each dimension (for a Gaussian density function) with corresponding standard error.

of smooth functions and  $m \in \mathcal{F}_{n,p,\delta}$  where

$$\mathcal{F}_{n,p,\delta} = \left\{ m : \sup_{x \in \mathcal{X}} \left| \frac{b(x)}{\sqrt{\hat{V}(x)}} \right| \leq \delta \right\},$$

then bands of the form

$$\left( \hat{m}_n(x) - (\delta + c)\sqrt{\hat{V}(x)}, \hat{m}_n(x) + (\delta + c)\sqrt{\hat{V}(x)} \right) \tag{7.20}$$

are a confidence band for  $m(x)$ , where the bias  $b(x)$  can be estimated using (7.10). Note that the confidence interval (7.20) expands the bands in the presence of bias rather than recentering the bands to allow for bias (7.19). Sun and Loader (1994) showed that, if  $c$  is chosen according (7.17), bands of the form (7.20) lead to a lower bound for the true coverage probability of the form

$$\inf_{m \in \mathcal{F}_{n,p,\delta}} \mathbf{P} \left\{ |\hat{m}_n(x) - m(x)| \leq c\sqrt{\hat{V}(x)}, \forall x \in \mathcal{X} \right\} = 1 - \alpha - O(\delta)$$

as  $\delta \rightarrow 0$ . The error term can be improved to  $O(\delta^2)$  if one considers classes of functions with bounded derivatives.

Recently, Krivobokova et al. (2010) have applied the volume-of-tube formula to penalized spline estimators. They showed that the mixed-model formulation of penalized splines can help to obtain, at least approximately, confidence bands with either Bayesian or frequentist properties. Further, they showed that for confidence bands, based on the conditional mixed model, the critical value for the width of the bands automatically accounts for the bias.

We conclude by summarizing the construction of simultaneous confidence intervals given in Algorithm 7.

---

**Algorithm 7** Construction of Simultaneous Confidence Bands

---

- 1: Given the data  $\{(X_1, Y_1), \dots, (X_n, Y_n)\}$ , calculate  $\hat{m}_n$  using (2.16)
- 2: Calculate residuals  $\hat{e}_k = Y_k - \hat{m}_n(X_k), k = 1, \dots, n$
- 3: Calculate the variance of the LS-SVM by using (7.11) and (7.13)
- 4: Calculate the bias using double smoothing (7.10)
- 5: Set significance level e.g.  $\alpha = 0.05$
- 6: Calculate the width  $c$  of the bands from (7.17)
- 7: Use (7.20) to obtain simultaneous confidence bands.

---

### 7.3.3 Pointwise and Simultaneous Prediction Intervals

In some cases one may also be interested in the uncertainty on the prediction for a new observation. This type of requirement is fulfilled by the construction of a prediction interval. Assume that the new observation  $Y^*$  at a point  $x^*$  is independent of the estimation data, then

$$\begin{aligned} \text{Var}[Y^* - \hat{m}_n(x^*)|X = x^*] &= \text{Var}[Y^*|X = x^*] + \text{Var}[\hat{m}_n(x^*)|X = x^*] \\ &= \sigma^2(x^*) + \sum_{i=1}^n l_i(x^*)^2 \sigma^2(x_i). \end{aligned}$$

Thus, an approximate pointwise  $100(1 - \alpha)\%$  prediction interval in a new point  $x^*$  is constructed by

$$\hat{m}_n(x^*) - \hat{b}(x^*) \pm z_{1-\alpha/2} \sqrt{\hat{\sigma}^2(x^*) + \hat{V}(x^*)}. \tag{7.21}$$

An approximate simultaneous  $100(1 - \alpha)\%$  prediction interval in a new point  $x^*$  is given by

$$\left( \hat{m}_n(x^*) - (\delta + c) \sqrt{\hat{\sigma}^2(x^*) + \hat{V}(x^*)}, \hat{m}_n(x^*) + (\delta + c) \sqrt{\hat{\sigma}^2(x^*) + \hat{V}(x^*)} \right). \tag{7.22}$$

## 7.4 Bootstrap Based Confidence and Prediction Intervals

In this Section we will briefly review the current state-of-the-art regarding bootstrap based confidence and prediction intervals, which are used for comparison in the experimental section.

### 7.4.1 Bootstrap Based on Residuals

Härdle (1989) showed that the standard bootstrap (Efron, 1979) based on residuals does not work for nonparametric heteroscedastic regression models. A technique used to overcome this difficulty is the wild or external bootstrap, developed in Liu (1988) following suggestions in Wu (1986) and Beran (1986). Further theoretical refinements are found in Mammen (1993). Algorithm 8 and Algorithm 9 have to be applied when the errors are homoscedastic and heteroscedastic respectively.

---

#### Algorithm 8 Bootstrap based on residuals (homoscedastic case)

---

- 1: Given the data  $\{(X_1, Y_1), \dots, (X_n, Y_n)\}$ , calculate  $\hat{m}_n$  using (2.16)
  - 2: Calculate residuals  $\hat{e}_k = Y_k - \hat{m}_n(X_k)$
  - 3: Re-center residuals  $\tilde{e}_k = \hat{e}_k - n^{-1} \sum_{j=1}^n \hat{e}_j$
  - 4: Generate bootstrap samples  $\{\tilde{e}_k^*\}_{k=1}^n$  by sampling with replacement from  $\{\tilde{e}_k\}_{k=1}^n$
  - 5: Generate  $\{Y_k^*\}_{k=1}^n$  from  $Y_k^* = \hat{m}_n(X_k) + \tilde{e}_k^*$
  - 6: Calculate  $\hat{m}_n^*$  from  $\{(X_1, Y_1^*), \dots, (X_n, Y_n^*)\}$
  - 7: Repeat steps (4)-(6)  $B$  times
- 

---

#### Algorithm 9 Bootstrap based on residuals (heteroscedastic case)

---

- 1: Given the data  $\{(X_1, Y_1), \dots, (X_n, Y_n)\}$ , calculate  $\hat{m}_n$  using (2.16)
  - 2: Calculate residuals  $\hat{e}_k = Y_k - \hat{m}_n(X_k)$
  - 3: Re-center residuals  $\tilde{e}_k = \hat{e}_k - n^{-1} \sum_{j=1}^n \hat{e}_j$
  - 4: Generate bootstrap data  $\tilde{e}_k^* = \tilde{e}_k \eta_k$  where  $\eta_k$  are Rademacher variables, defined as
 
$$\eta_k = \begin{cases} 1, & \text{with probability } 1/2; \\ -1, & \text{with probability } 1/2. \end{cases}$$
  - 5: Generate  $\{Y_k^*\}_{k=1}^n$  from  $Y_k^* = \hat{m}_n(X_k) + \tilde{e}_k^*$
  - 6: Calculate  $\hat{m}_n^*$  from  $\{(X_1, Y_1^*), \dots, (X_n, Y_n^*)\}$
  - 7: Repeat steps (4)-(6)  $B$  times
- 

Other possibilities for the two-point distribution in Algorithm 9 of the  $\eta_k$  also exist, see e.g. Liu (1988). The Rademacher distribution was chosen because it was empirically shown in Davidson et al. (2007) that this distribution came out as best among six alternatives.

### 7.4.2 Construction of Bootstrap Confidence and Prediction Intervals

The construction of bootstrap confidence and prediction intervals for nonparametric function estimation consists of two parts, i.e. the construction of a confidence



or a prediction interval based on a pivotal method for the expected value of the estimator and bias correction through undersmoothing. Then, a confidence interval is constructed by using the asymptotic distribution of a pivotal statistic. The latter can be obtained by bootstrap. Before illustrating the construction of intervals based on bootstrap, we give a formal definition of a key quantity used in the bootstrap approach.

**Definition 7.2 (Pivotal quantity)** *Let  $X = (X_1, \dots, X_n)$  be random variables with unknown joint distribution  $F$  and denote by  $T(F)$  a real-valued parameter of interest (e.g. the regression function). A random variable  $\mathcal{T}(X, T(F))$  is a pivotal quantity (or pivot) if the distribution of  $\mathcal{T}(X, T(F))$  is independent of all parameters.*

Hall (1992) suggested the following approach: estimate the distribution of the pivot

$$\mathcal{T}(m(x), \hat{m}_n(x)) = \frac{\hat{m}_n(x) - m(x)}{\sqrt{\hat{V}(x)}}$$

by the bootstrap. Depending on homoscedastic or heteroscedastic errors Algorithm 8 or Algorithm 9 should be used to estimate this distribution. Now, the distribution of the pivotal statistic  $\mathcal{T}(m(x), \hat{m}_n(x))$  is approximated by the corresponding distribution of the bootstrapped statistic

$$\mathcal{V}(\hat{m}_n^*(x), \hat{m}_{n,g}(x)) = \frac{\hat{m}_n^*(x) - \hat{m}_{n,g}(x)}{\sqrt{\hat{V}(x)}}$$

where  $\hat{m}_{n,g}$  denotes the undersmoother with bandwidth  $g$  and  $\star$  denote bootstrap counterparts. Practically, we choose bandwidth  $g = h/3$ . Hence, a  $100(1 - \alpha)\%$  pointwise confidence interval is given by

$$(\Psi_{\alpha/2}, \Psi_{1-\alpha/2}),$$

with

$$\Psi_{\alpha/2} = \hat{m}_n(x) + Q_{\alpha/2} \sqrt{\hat{V}(x)}$$

and

$$\Psi_{1-\alpha/2} = \hat{m}_n(x) + Q_{1-\alpha/2} \sqrt{\hat{V}(x)}.$$

$Q_\alpha$  denotes the  $\alpha^{\text{th}}$  quantile of the bootstrap distribution of the pivotal statistic. A  $100(1 - \alpha)\%$  simultaneous confidence interval can be constructed by applying a Šidák correction or using the length heuristic (Efron, 1997). Similarly,  $100(1 - \alpha)\%$  pointwise and simultaneous prediction intervals are obtained.

A question which remains unanswered is how to determine  $B$ , the number of bootstrap replications in Algorithm 8 and Algorithm 9. The construction of

confidence (and prediction) intervals demands accurate information of the low and high quantiles of the limit distribution. Therefore, enough resamplings are needed in order for bootstrap to accurately reproduce this distribution. Typically,  $B$  is chosen in the range of 1.000-2.000 for pointwise intervals and more than 10.000 for simultaneous intervals.

## 7.5 Simulations: The Regression Case

In all simulations, the Gaussian kernel was used and  $\alpha = 0.05$ . The tuning parameters (regularization parameter  $\gamma$  and kernel bandwidth  $h$ ) of the LS-SVM were obtained via leave-one-out cross-validation.

### 7.5.1 Empirical Coverage Probability

To assess the performance of the proposed simultaneous CI in practice (for the homoscedastic case), we conducted a Monte Carlo experiment. Data was generated from model (7.1) using normal errors and for the regression curve

$$m(x) = e^{-32(x-0.5)^2}.$$

Sample sizes  $n = 50, 100, 200, 400, 600$  and  $800$  were used and  $\sigma(x) = \sigma$  was chosen to be  $0.05$  and  $0.1$ . We investigated simultaneous confidence intervals (7.20) for  $\alpha = 0.05$ . For this experiment we generated 5000 replicate samples for each setting (using a different seed in each setting). We recorded the proportion of times all the  $m(x_k)$ ,  $k = 1, \dots, n$ , fell inside the bands. In addition, empirical coverage probabilities were computed for the proposed simultaneous intervals without bias correction, pointwise bias-corrected intervals (7.15), Šidák confidence bands with and without bias correction, i.e.

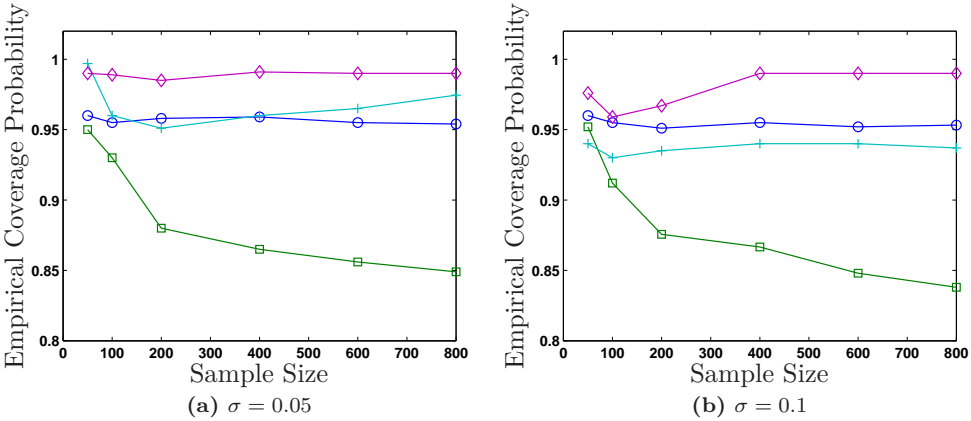
$$\hat{m}_n(x) - \hat{b}(x) \pm z_{1-\beta/2} \sqrt{\hat{V}(x)}$$

and

$$\hat{m}_n(x) \pm z_{1-\beta/2} \sqrt{\hat{V}(x)}$$

respectively where  $\beta = 1 - (1 - \alpha)^{1/n}$ . Figure 7.5 shows the empirical coverage probabilities for the different methods. For illustration purposes, we did not include the pointwise bias-corrected intervals since their coverage probability is too low (around 0.25), which is expected. We observe that the proposed simultaneous CIs attain the coverage probability close to the nominal value, i.e. 0.95, for both noise levels. We find that, if no bias correction is included in the proposed simultaneous CIs, the coverage probability is on average 5% to 10% lower than the

nominal value. These results confirm that a suitable bias correction is absolutely necessary to attain proper coverage and that centering the bands to allow for bias is not a good strategy. It is also clear that the Šidák confidence bands produce conservative bands, i.e. a larger coverage probability is attained for all the simulated sample sizes, even if no bias correction is performed for the low noise level. This is because the Šidák correction gives a very crude upperbound of the width of the intervals and the noise level is still small. However, increasing the noise level (see Figure 7.5b) also leads to undercoverage for these type of bands.



**Figure 7.5:** Empirical coverage probabilities for the proposed simultaneous CIs with bias correction (o), proposed simultaneous CIs without bias correction (□), Šidák confidence bands with bias correction (+), Šidák confidence bands without bias correction (◇).

### 7.5.2 Homoscedastic Examples

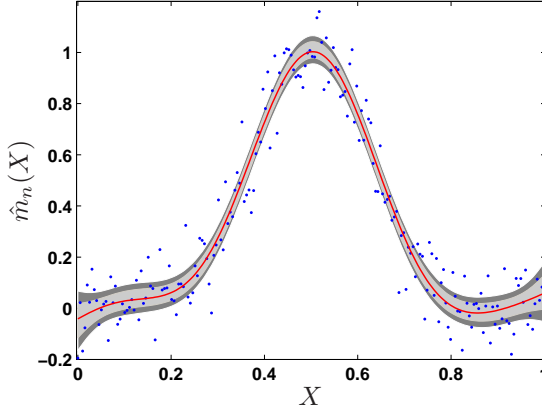
In the first example, data were generated from model (7.1) using normal errors and following the regression curve

$$m(x) = e^{-32(x-0.5)^2}.$$

The sample size is taken to be  $n = 200$  and  $\sigma^2(x) = \sigma^2 = 0.01$ . Pointwise (7.15) and simultaneous 95% confidence intervals (7.20) are shown in Figure 7.6. The line in the middle represents the LS-SVM model. For illustration purposes the 95% pointwise confidence intervals are connected.

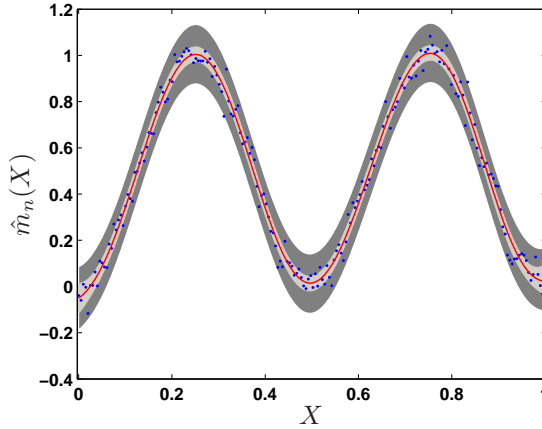
In a second example, we generate data from model (7.1) following the regression curve (normal errors with  $\sigma^2(x) = \sigma^2 = 0.01$ )

$$m(x) = \sin^2(2\pi x).$$



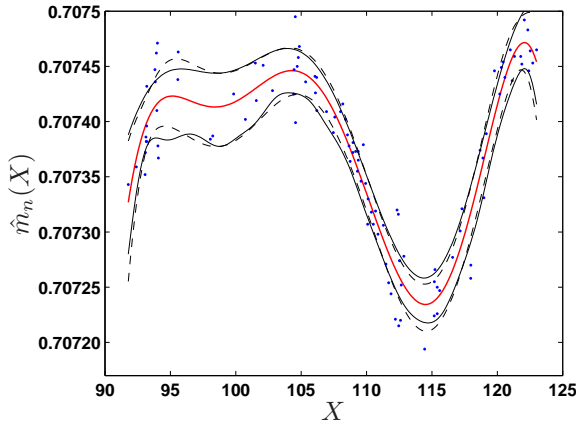
**Figure 7.6:** Pointwise and simultaneous 95% confidence intervals. The outer (inner) region corresponds to simultaneous (pointwise) confidence intervals. The full line (in the middle) is the estimated LS-SVM model. For illustration purposes the 95% pointwise confidence intervals are connected.

Figure 7.7 illustrates the 95% simultaneous confidence (7.20) and prediction intervals (7.22). The outer (inner) region corresponds to the prediction (confidence) interval.



**Figure 7.7:** Simultaneous 95% confidence and prediction intervals. The outer (inner) region corresponds to simultaneous prediction (confidence) intervals. The full line (in the middle) is the estimated LS-SVM model.

In a third example, we compare the proposed simultaneous confidence intervals (7.20) with the bootstrap method on the Fossil data set. The number of bootstrap replications was  $B = 15,000$ . From Figure 7.8 it is clear that both methods produce similar confidence intervals. It is shown that our proposed CIs produce intervals which are close to the ones obtained by bootstrap.



**Figure 7.8:** Simultaneous 95% confidence intervals for the Fossil data set. The dashed lines correspond to the proposed simultaneous confidence intervals and the full lines are the bootstrap confidence intervals. The full line (in the middle) is the estimated LS-SVM model.

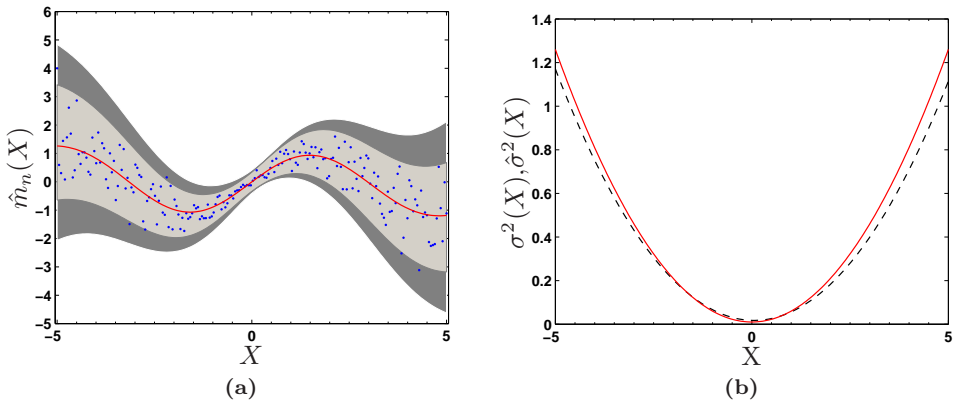
### 7.5.3 Heteroscedastic Examples and Error Variance Estimation

The data is generated according to the following model (with standard normal errors)

$$Y_k = \sin(x_k) + \sqrt{0.05x_k^2 + 0.01} e_k, \quad k = 1, \dots, 200,$$

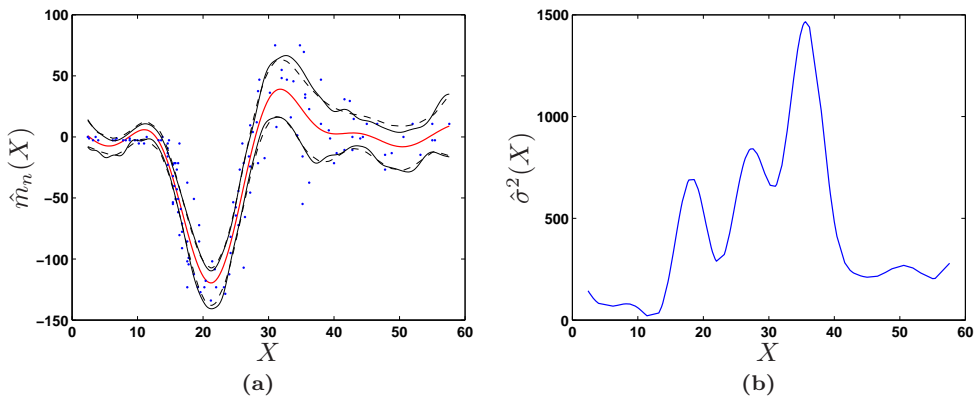
where the  $x_k$  are equally spaced over the interval  $[-5, 5]$  and  $e \sim \mathcal{N}(0, 1)$ . Figure 7.9a and Figure 7.9b show 95% pointwise (7.21) and simultaneous (7.22) prediction intervals for this model and the estimated (and true) variance function respectively. The variance function was obtained by Theorem 7.3. The latter clearly demonstrates the capability of the proposed methodology for variance estimation.

As a last example, consider the Motorcycle data set. We compare the proposed simultaneous confidence intervals (7.20) with the wild bootstrap method. The number of bootstrap replications was  $B = 15,000$ . The result is given in



**Figure 7.9:** (a) Pointwise and simultaneous 95% prediction intervals for heteroscedastic errors. The outer (inner) region corresponds to simultaneous (pointwise) prediction intervals. The full line (in the middle) is the estimated LS-SVM model. For illustration purposes the 95% pointwise prediction intervals are connected; (b) Variance function estimation. The full line represents the real variance function and the dashed line is the estimated variance function obtained by Theorem 7.3.

Figure 7.10a. As before, both intervals are very close to each other. Figure 7.10b shows the estimated variance function  $\hat{\sigma}^2(\cdot)$  of this data set.



**Figure 7.10:** (a) Simultaneous 95% confidence intervals for the Motorcycle data set. The dashed lines correspond to the proposed simultaneous confidence intervals and the full lines are the bootstrap confidence intervals. The full line (in the middle) is the estimated LS-SVM model; (b) Variance function estimation of the Motorcycle data set obtained by Theorem 7.3.

## 7.6 Confidence Intervals: Classification

### 7.6.1 Classification vs. Regression

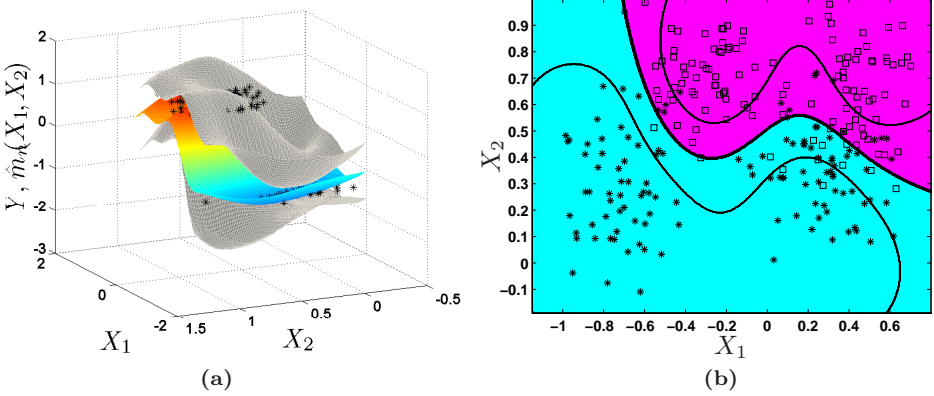
Given a training set defined as  $\mathcal{D}_n = \{(X_k, Y_k) : X_k \in \mathbb{R}^d, Y_k \in \{-1, +1\}; k = 1, \dots, n\}$ , where  $X_k$  is the  $k^{\text{th}}$  input pattern and  $Y_k$  is the  $k^{\text{th}}$  output pattern. In the primal weight space, LS-SVM for classification is formulated as ( Suykens and Vandewalle, 1999)

$$\begin{aligned} \min_{w,b,e} \mathcal{J}_c(w,e) &= \frac{1}{2}w^T w + \frac{\gamma}{2} \sum_{i=1}^n e_i^2 \\ \text{s.t. } Y_i[w^T \varphi(X_i) + b] &= 1 - e_i, \quad i = 1, \dots, n, \end{aligned} \quad (7.23)$$

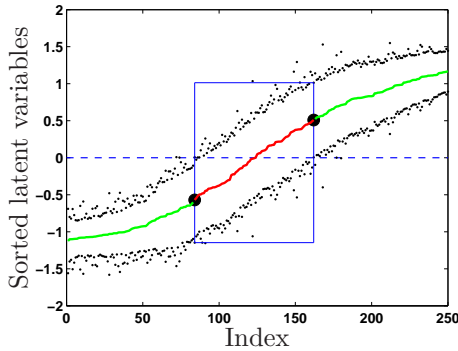
However, by using the substitution  $e_i = Y_i e_i$  in (7.23), which does not change the objective function since  $Y_i^2 = 1$ , LS-SVM formulations for classification and regression are equivalent. Therefore, we can also apply all of the above techniques for constructing CIs in the classification case, by considering the classification problem as a regression problem.

### 7.6.2 Illustration and Interpretation of the Method

We will graphically illustrate the proposed method for the construction of CIs on the Ripley data set. First, an LS-SVM regression model from the Ripley data set is estimated according to (2.16). From the obtained model, confidence bands can be calculated using (7.20) and based on the two dimensional volume-of-tube formula (7.16). Figure 7.11 shows the obtained results in three dimensions and its two dimensional projection respectively. In the latter the two outer bands represent 95% confidence intervals for the classifier. An interpretation of this result can be given as follows. For every point within or without the two outer bands, the classifier casts doubt with significance level  $\alpha$  or is confident with significance level  $\alpha$  on its label respectively. However, in higher dimensions, the previous figures cannot be made anymore. Therefore, we can visualize the classifier via its latent variables, i.e. the output of the LS-SVM before taking the sign function, and show the corresponding confidence intervals, see Figure 7.12. The middle full line represents the sorted latent variables of the classifier. The dots above and below the full line are the 95% confidence intervals of the classifier. These dots correspond to the confidence bands in Figure 7.11. The dashed line at zero represents the decision boundary. The rectangle visualizes the critical area for the latent variables. Hence, for all points with latent variables between the two big dots, the classifier, i.e.  $|\text{latent variable}| \leq 0.51$ , casts doubt on the corresponding label with significance level of 5%. Such a visualization can always be made and can assist the user in assessing the quality of the classifier.



**Figure 7.11:** Ripley data set (a) Regression on the Ripley data set with corresponding 95% confidence intervals obtained with (7.20) where  $X_1, X_2$  are the corresponding abscissa and  $\hat{m}_n(X_1, X_2)$  the function value; (b) Two dimensional projection of (a) obtained by cross-sectioning the regression surfaces with the decision plane  $Y = 0$ . The two outer lines represent 95% confidence intervals on the classifier. The line in the middle is the resulting classifier.

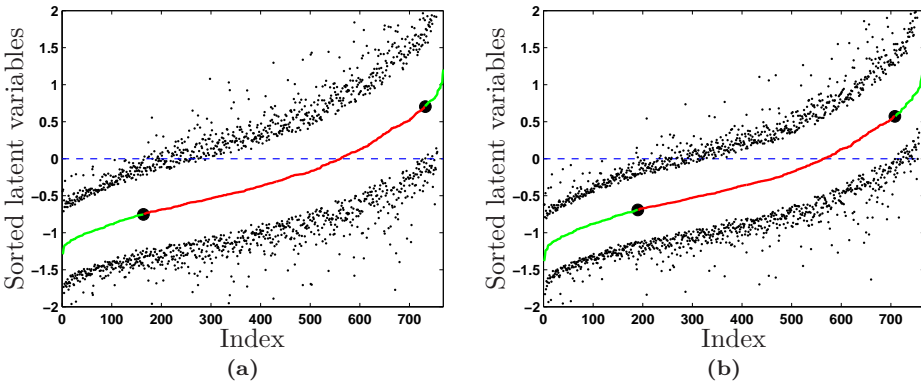


**Figure 7.12:** Visualization of the 95% confidence bands (small dots above and below the middle line) for the Ripley data set based on the latent labels (middle full line). The rectangle visualizes the critical area for the latent variables, therefore, every point lying between the two big dots the classifier casts doubt on its label with significance level of 5%. The dashed line is the decision boundary.



### 7.6.3 Simulations: The Classification Case

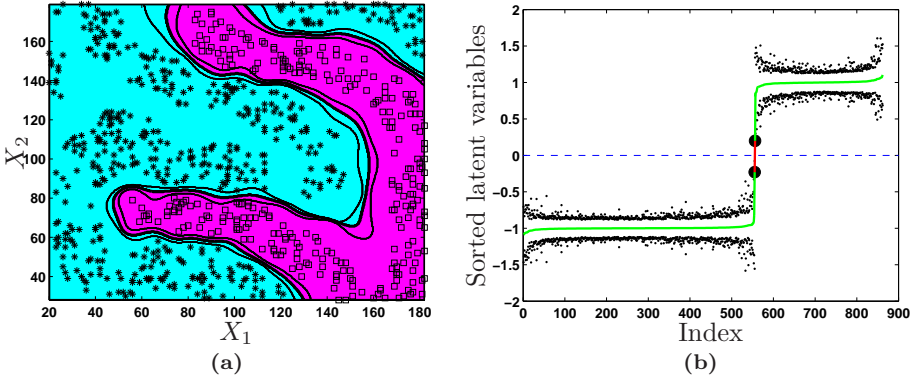
First, consider the Pima Indians data set ( $d = 8$ ). Figure 7.13 shows the  $100(1 - \alpha)\%$  confidence bands for the classifier based on the latent variables where  $\alpha$  is varied from 0.05 to 0.1 respectively. Figure 7.13a illustrates the 95% confidence bands for the classifier based on the latent variables and Figure 7.13b the 90% confidence bands. It is clear that the width of the confidence band will decrease when  $\alpha$  increases. Hence, the 95% and 90% confidence band for the latent variables is given roughly by  $(-0.70, 0.70)$  and  $(-0.53, 0.53)$ . Second, consider the Fourclass



**Figure 7.13:** Pima Indians data set. Effect of larger significance level on the width of the confidence bands. The bands will become wider when the significance level decreases. (a) 95% confidence band on the latent variables; (b) 90% confidence band on the latent variables.

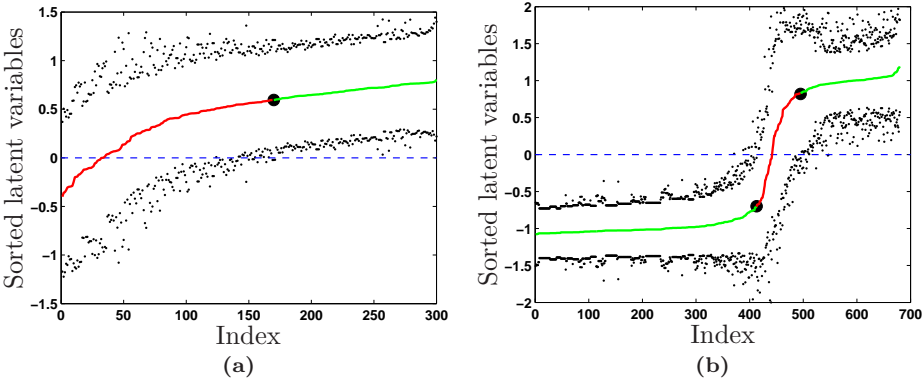
data set ( $d = 2$ ). This is an example of a non-linear separable classification problem (see Figure 7.14a). We can clearly see in Figure 7.14a that the 95% confidence bands are not wide, indicating no overlap between classes. Figure 7.14b shows the 95% confidence bands for the classifier based on the latent variables. The two black dots indicate the critical region. Therefore, if for any point  $|\text{latent variable}| \leq 0.2$  we have less than 95% confidence on the decision of the class label.

As a third example, consider the Haberman’s Survival data set ( $d = 3$ ). The data set contains cases from a study that was conducted between 1958 and 1970 at the University of Chicago’s Billings Hospital on the survival of patients who had undergone surgery for breast cancer. The task is to predict whether the patient will survive five years or longer. Figure 7.15a shows the 95% confidence bands for the latent variables. Every point lying left from the big dot i.e. latent variable  $< 0.59$ , the classifier casts doubt on its label on a significance level of 5%. This is a quite difficult classification task as can be seen from the elongated form of the sorted latent variables and is also due to the unbalancedness in the data.



**Figure 7.14:** Fourclass data set. (a) Two dimensional projection of the classifier (inner line) and its corresponding 95% confidence bands (two outer lines); (b) Visualization of the 95% confidence bands for classification based on the latent labels (middle full line). For every latent variable lying between the two big dots the classifier casts doubt on its label. The dashed line is the decision boundary.

A fourth example is the Wisconsin Breast Cancer data set ( $d = 10$ ). Figure 7.15b shows the 95% confidence bands for the latent variables. Thus every point lying between the big dots, the classifier casts doubt on its label on a significance level of 5%.



**Figure 7.15:** (a) Visualization of the 95% confidence bands for Haberman’s survival data set based on the latent labels (middle full line). For every latent variable lying left from the big dot the classifier casts doubt on its label on a 5% significance level. The dashed line is the decision boundary; (b) Visualization of the 95% confidence bands for Wisconsin Breast Cancer data set based on the latent labels. For every latent variable lying between the big dots the classifier casts doubt on its label on a 5% significance level. The dashed line is the decision boundary.

## 7.7 Conclusions

In this Chapter, we discussed the construction of bias-corrected  $100(1 - \alpha)\%$  approximate confidence and prediction intervals (pointwise and simultaneous) for linear smoothers, in particular for LS-SVM. Under certain conditions, we proved the asymptotic normality of LS-SVM. We discussed a technique called double smoothing to determine the bias without estimating higher order derivatives. Further, we developed a nonparametric variance estimator which can be related to other well-known nonparametric variance estimators. In order to obtain uniform or simultaneous confidence intervals we used two techniques i.e Bonferroni/Šidák correction and volume-of-tube formula. We provided extensions of this formula in higher dimensions and discussed how to compute some of the coefficients in practice. We illustrated that the width of the bands are expanding with increasing dimensionality by means of an example. By means of a Monte Carlo study, we demonstrated that the proposed bias-corrected  $100(1 - \alpha)\%$  simultaneous confidence intervals achieve the proper empirical coverage rate. Finally, the results for the regression case are extended to the classification case.



# Chapter 8

## Applications and Case Studies

In this Chapter, we will discuss some practical examples and case studies. We will demonstrate the capabilities of the developed techniques on system identification, density estimation and on finding maxima in hysteresis curves.

### 8.1 System Identification with LS-SVMLab

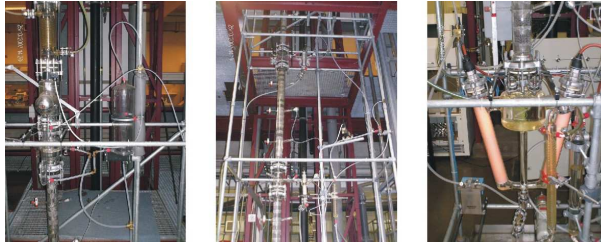
#### 8.1.1 General Information

In this Section we describe how the user can identify a system based on input-output data via the LS-SVMLab v1.7 toolbox (De Brabanter et al., 2010b). As an example, consider the pilot scale distillation column data set (Huyck et al., 2010). One of the key task is to model the bottom temperature of the column given the following inputs: feed flow rate, feed duty, reboiler duty and reflux flow rate. Photos of the distillation column are shown in Figure 8.1. The training data set consists out of the first 2/3 of the recorded data set and the remaining 1/3 is taken as validation data. Let  $U$  and  $Y$  denote the input matrix and output vector of the complete data set. The data has zero mean and unit variance.

#### 8.1.2 Model Identification

We consider the following model structure (nonlinear ARX)

$$y_t = f(y_{t-1}, \dots, y_{t-p}; u_{t-1}, \dots, u_{t-p}) + e_t,$$



**Figure 8.1:** Distillation column (left) condenser; (center) packed section and feed introduction; (right) reboiler. (Courtesy of Bart Huyck)

where  $p$  denotes the order of the NARX model (number of lags),  $u$  and  $y$  denote the inputs and outputs of the system respectively. The number of lags are determined via the MSE on validation data. The following code can be employed to determine the MSE for a certain lag  $p$  on validation data with the LS-SVMLab software (using an RBF kernel).

```
% Re-arrange the data into a block Hankel matrix for NARX
% Data points 1 to R contain the training data
>> [Xw,Yw]=windowizeNARX(U(1:R,:),Y(1:R),p,p);
>> Ytraininit=Xw(1,end-p+1:end); Xwtrain = Xw(:,1:end-p);

% Train and tune the model in a fully automatic way (10-fold CV)
>> model=initlssvm(Xw,Yw,'f',[[]],[],'RBF_kernel');
>> model=tunelssvm(model,'simplex','crossvalidatelssvm',{10,'mse'});
>> model=trainlssvm(model);

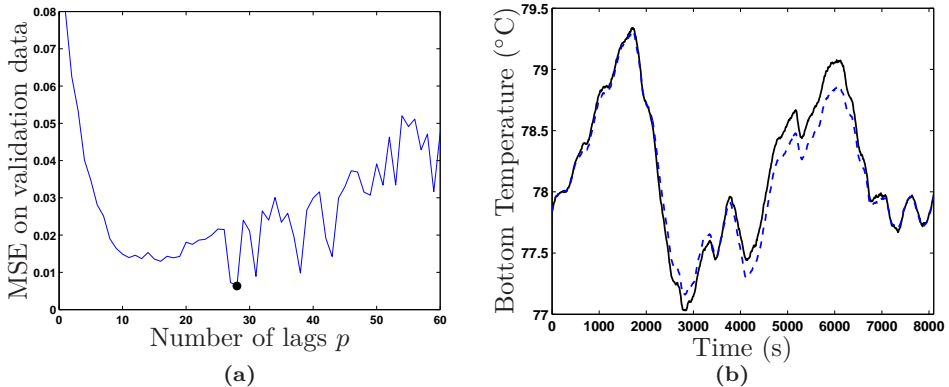
% Create block Hankel matrix to simulate model on validation
>> [Xweval,~]=windowizeNARX(U(R+1:end,:),Y(R+1:end),p,p);

% Set starting point for simulation
>> Yinit=Xweval(1,end-p+1:end); Xweval=Xweval(:,1:end-p);

% Start simulation at zero
>> sim=zeros(1,size(Xweval,1)+p);
>> sim(p+1-(1:p)')=fliplr(Yinit);
>> for n=1:size(Xweval,1)
>> sim(1,p+n)=simlssvm(model,[Xweval(n,:) fliplr(sim(p+n-(1:p)')]);
>> end
>> sim=sim';

% MSE on validation data
>> errorL2 = mse(Y(R+1:end)-sim)
```

Using the above code for various lags  $p$  e.g. from 1 to 60, we obtain the MSE on validation data as a function of the number of lags  $p$ . From Figure 8.2a it is clear the number of lags are set to  $p = 28$ . Finally, setting  $p = 28$  and train the final LS-SVM will give the result on test data. The final result is illustrated in Figure 8.2b. The MSE on test data is 0.0063.



**Figure 8.2:** (a) MSE on validation data as a function of the number of lags  $p$ . The dot ( $p = 28$ ) indicates the lowest value of the MSE on validation data.; (b) Final LS-SVM model (dotted line) with  $p = 28$  on test data. The full line represents the measured data. (Courtesy of Bart Huyck)

## 8.2 SYSID 2009: Wiener-Hammerstein Benchmark

### 8.2.1 Model Structure

Since this was a large-scale problem, we advocate the use of FS-LSSVM (see Chapter 4). For this method different number of prototype vectors are selected. All subsamples are selected by maximizing the quadratic Rényi entropy criterion, see Chapter 4 Algorithm 1.

The general model structure is a nonlinear ARX (NARX) of the form

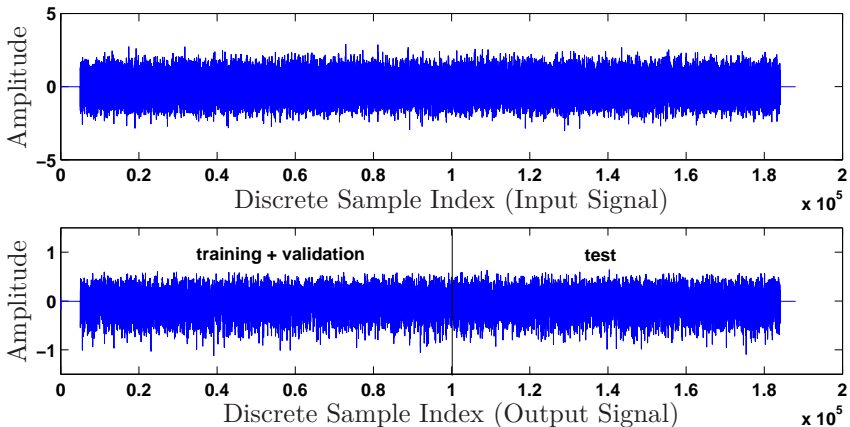
$$y_t = f(y_{t-1}, \dots, y_{t-p}; u_{t-1}, \dots, u_{t-p}) + e_t,$$

where  $p$  denotes the order of the NARX model (number of lags). The number of lags are determined via 10-fold cross-validation.

## 8.2.2 Data Description and Training Procedure

The data consists of samples for the input  $u_i$  and the outputs  $y_i$ , with  $i = 1, \dots, 188.000$ . A plot of the inputs and outputs is given in Figure 8.3. Next, the strategy for the using the data in terms of training and testing will be outlined. This goes as follows:

- Training + validation sample: from data point 1 to data point 100.000. Using 10-fold CV, models are repeatedly (10 times) estimated using 90% of the data and validated on the remaining 10%. Two approaches will be used here i.e CV on a one-step-ahead-basis ( $\text{CV-RMSE}_1$ ) and CV based on simulating the estimated model ( $\text{CV-RMSE}_{\text{sim}}$ ). The mean squared error (MSE) for a one-step-ahead prediction/simulation can be computed using this validation sample. The number of lags  $p$  are determined by the lowest value of the MSE of the cross-validation function.
- Test sample: from data points 100.001 to data points 188.000. After defining the optimal lags  $p$  and optimal tuning parameters  $\gamma$  and  $\sigma$  (in case of RBF kernel), the prediction on the test set can be done. In this paper, an iterative prediction is computed for the entire test set. This is done by using each time the past predictions as inputs while using the estimated model in simulation mode.



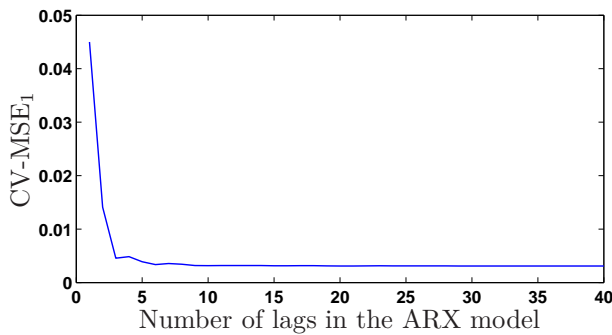
**Figure 8.3:** Available data for the Wiener-Hammerstein identification problem. The zones for training + validation (estimation set) and test are indicated in the output series.



### 8.2.3 Estimation and Model Selection

Using the above described training + validation scheme (10-fold CV), we start checking different lag orders and tuning parameters. Each time the model is repeatedly estimated using the training set (90% of the training data) and validated using the remaining 10%. This is done on a one-step-ahead basis and simulation basis. The best model is selected based on the lowest MSE on cross-validation ( $CV-MSE_1$  or  $CV-MSE_{sim}$ ).

Consider a linear ARX with varying input and output lags. The model order is determined by 10-fold CV ( $CV-MSE_1$ ). Figure 8.4 shows the  $CV-MSE_1$  obtained for lags varying from 1 to 40.



**Figure 8.4:** The error on cross-validation ( $CV-MSE_1$ ) using a linear ARX model with increasing number of lags.

Table 8.1 shows the best results in RMSE on cross-validation (one-step-ahead based ( $CV-RMSE_1$ ) and simulation based  $CV-RMSE_{sim}$ ) obtained for each of the techniques. NARX is a nonlinear ARX model obtained with the Matlab System Identification Toolbox. The lags for MLP-NARX are found by validation on a single set. The nonlinearity was modeled with an MLP and a sigmoid activation function with 10 hidden neurons. Due to the use of a single validation set the lags for MLP-NARX differ from the rest. For the FS-LSSVM three kernel types are reported i.e. RBF, polynomial and linear. All three techniques make use of the complete training data set of 100.000 data points. All RMSE figures are expressed in the original units of the data.

From the results in Table 8.1, it is clear that the FS-LSSVM using the RBF kernel outperforms the others. The linear ARX is unable to capture the nonlinearity in the data resulting in a lower performance on cross-validation RMSE (up to 2 orders of magnitude). Although two nonlinear techniques (NARX and FS-LSSVM) are used, their performances are quite different.

The effect of varying numbers of selected prototype vectors  $m$  on both cross-validation techniques are reported in Table 8.2 for the FS-LSSVM with RBF kernel. The best performance is bold faced in Table 8.2. The total number of prototype vectors is set to  $m = 5000$ . The position of the selected prototype vectors (quadratic Rényi entropy criterion) is shown according to the corresponding position of the input data. Figure 8.5 shows the training input data together with the position of the selected prototype vectors, represented by dark bars.

**Table 8.1:** Best models based on cross-validation RMSE. MLP-NARX is a nonlinear ARX model obtained with the Matlab SYSID Toolbox. Two types of CV are displayed: CV based on one-step-ahead (CV-RMSE<sub>1</sub>) and simulation (CV-RMSE<sub>sim</sub>)

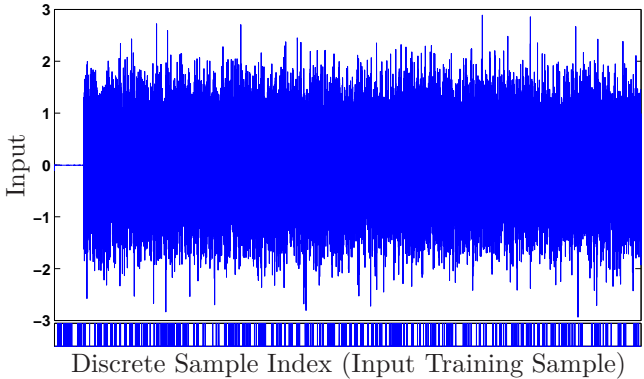
Method	Kernel	lags	CV-RMSE <sub>1</sub>	CV-RMSE <sub>sim</sub>
ARX	-	10	$5.67 \times 10^{-2}$	$5.66 \times 10^{-2}$
MLP-NARX	-	11	$7.62 \times 10^{-4}$	$2.15 \times 10^{-2}$
FS-LSSVM	Lin	10	$8.64 \times 10^{-4}$	$4.51 \times 10^{-2}$
	Poly	10	$5.63 \times 10^{-4}$	$5.87 \times 10^{-3}$
	RBF	10	$4.77 \times 10^{-4}$	$4.81 \times 10^{-3}$

**Table 8.2:** Effect of different numbers of prototype vectors  $m$  on the performance (CV-RMSE<sub>1</sub> and CV-RMSE<sub>sim</sub>) of the FS-LSSVM estimator with RBF kernel. The chosen number of prototype vectors is bold faced.

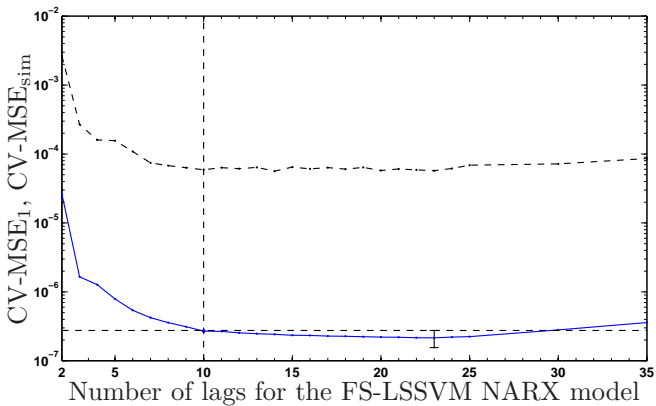
$m$	CV-RMSE <sub>1</sub>	CV-RMSE <sub>sim</sub>
100	$5.82 \times 10^{-4}$	$2.14 \times 10^{-2}$
400	$5.36 \times 10^{-4}$	$7.85 \times 10^{-3}$
600	$5.13 \times 10^{-4}$	$6.76 \times 10^{-3}$
800	$5.05 \times 10^{-4}$	$5.87 \times 10^{-3}$
1200	$4.95 \times 10^{-4}$	$5.52 \times 10^{-3}$
1500	$4.93 \times 10^{-4}$	$5.05 \times 10^{-3}$
1750	$4.91 \times 10^{-4}$	$5.01 \times 10^{-3}$
2000	$4.89 \times 10^{-4}$	$4.98 \times 10^{-3}$
2500	$4.88 \times 10^{-4}$	$4.97 \times 10^{-3}$
<b>5000</b>	<b><math>4.77 \times 10^{-4}</math></b>	<b><math>4.81 \times 10^{-3}</math></b>

Finally, the effect of different lags was tested for lags varying from 2 to 35. Figure 8.6 shows the evolution of the lags on cross-validation MSE (one-step-ahead based and simulation based) for the model based on  $m = 5000$  (best model). In these experiments the number of input lags and output lags was kept equal to each

other (setting different input and outputs lags did not result in better performance on cross-validation MSE). For this example it did not matter whether the lags were selected by  $CV-MSE_1$  or  $CV-MSE_{sim}$ . The CV line only moves up and does not show any significant shifts to left or right. Thus, the selection of the number of lags seems independent of the chosen CV criterion. Selecting the number of lags is based on the least complex model that falls within one standard error (represented by the error bar at lag 23) of the best model (Hastie et al., 2009). In this case the number of lags have chosen to be 10 (vertical dashed line in Figure 8.6).



**Figure 8.5:** (top) The input training sample; (bottom) the position, as time index, of the 5000 selected prototype vectors by quadratic Rényi entropy is represented by the dark bars.



**Figure 8.6:** MSE on cross-validation ( $CV-MSE_1$  full line,  $CV-MSE_{sim}$  dash-dotted line) for FS-LSSVM NARX model ( $m = 5000$ ) for different number of lags. The number of lags are chosen so that the least complex model falls within one standard error (error bar at lag 23) of the best, i.e. number of lags equal to 10.

## 8.2.4 Results on Test Data

After selecting the model order and the involved parameters each of the models is used to make an iterative prediction i.e. using only past predictions and input information, for data points starting at sample 100.001 to sample 188.000. Since this is unseen data for the model, the following source of error can be expected: due to the iterative nature of the simulation, past errors can propagate to the next predictions. From the difference between the iterative prediction and the true values, the root mean squared error (RMSE) on test ( $\text{RMSE}_{\text{test}}$ ) is computed. In all results on test data the initial conditions for simulation were set to the real output values (first lag samples). The first 1001 samples of the prediction are omitted from consideration to eliminate the influence of transient errors.

Table 8.3 shows the performances ( $\text{RMSE}_{\text{test}}$ ) of the iterative prediction on test data for all types of model structures. The FS-LSSVM, using an RBF kernel, outperforms the ARX and NARX by a factor 10 and 4 respectively on RMSE on test data. The last column in Table 8.3 gives a fit percentage, i.e. the percentage of the output variation explained by the model and is defined as

$$\text{fit} = 100 \left( 1 - \frac{\|y - \hat{y}\|}{\|y - \bar{y}\|} \right),$$

where  $y$  is the measured output,  $\hat{y}$  the simulated output and  $\bar{y}$  the mean of  $y$ . Table 8.4 shows for the three different models the following results: the mean value of the iterative prediction and the standard deviation of the error (on test) defined as

$$\mu_t = \frac{1}{8700} \sum_{t=101001}^{188000} e(t) \quad \text{and} \quad s_t = \frac{1}{8700} \sum_{t=101001}^{188000} \sqrt{(e(t) - \mu_t)^2}$$

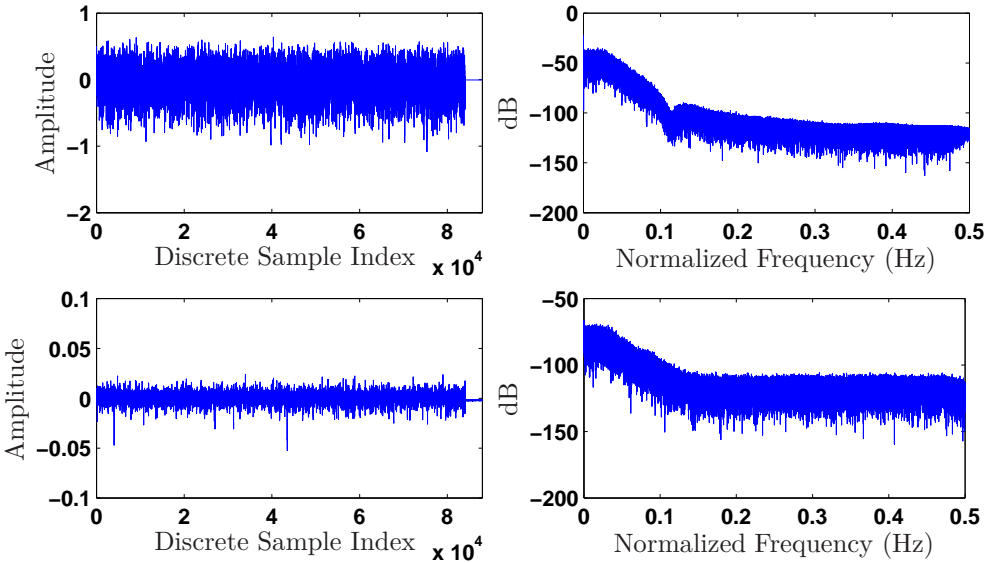
respectively, with  $e(t)$  the simulation error. Figure 8.7 shows the result of the final iterative prediction and the corresponding errors in the time and frequency domain.

**Table 8.3:** RMSE and fit percentage with the final iterative prediction using the model in simulation mode on the predefined test set.  $n_h$  denotes the number of hidden neurons in the MLP.  $n_h$  and lags for MLP-NARX are found by validation on a single set.

Method	lags/ $n_h$	$\text{RMSE}_{\text{test}}$	fit (%)
Linear ARX	10/-	$5.6 \times 10^{-2}$	76.47
MLP-NARX	11/15	$2.3 \times 10^{-2}$	86.06
FS-LSSVM (Lin)	10/-	$4.3 \times 10^{-2}$	81.93
FS-LSSVM (Poly)	10/-	$6.0 \times 10^{-3}$	96.86
FS-LSSVM (RBF)	10/-	$4.7 \times 10^{-3}$	97.98

**Table 8.4:** RMSE on training ( $RMSE_{tr}$ ) with iterative prediction on the training data using the model in simulation mode. The mean value of the simulation error  $\mu_t$  and the standard deviation of the error  $s_t$  are also reported (on test data).

Method	$\mu_t$	$s_t$	$RMSE_{tr}$
Linear ARX	$-3.6 \times 10^{-2}$	$4.3 \times 10^{-2}$	$5.5 \times 10^{-2}$
MLP-NARX	$-2.4 \times 10^{-3}$	$2.7 \times 10^{-2}$	$2.2 \times 10^{-2}$
FS-LSSVM (Lin)	$-1.4 \times 10^{-4}$	$4.3 \times 10^{-2}$	$4.2 \times 10^{-2}$
FS-LSSVM (Poly)	$1.2 \times 10^{-4}$	$6.1 \times 10^{-3}$	$5.9 \times 10^{-3}$
FS-LSSVM (RBF)	$6.3 \times 10^{-5}$	$4.8 \times 10^{-3}$	$4.5 \times 10^{-3}$



**Figure 8.7:** (top left) Iterative prediction (simulation mode) of the test data; (top right) Normalized frequency plot of the simulated test data; (bottom left) Errors of the iterative prediction (simulation mode) in the test set; (bottom right) Normalized frequency plot of the errors of the iterative prediction.

## 8.3 Nonparametric Comparison of Densities Based on Statistical Bootstrap

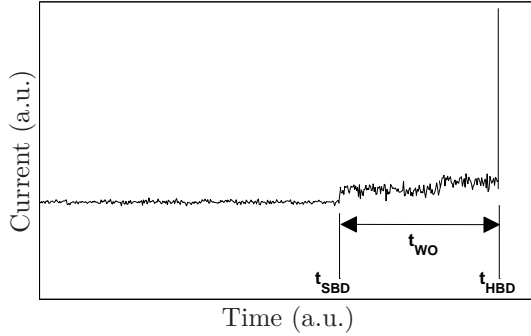
In this Section, we study which user specified parametric density best fits a given data set. It is quite common in the field of reliability analysis of MOSFETs to model failure times with a Weibull or a lognormal distribution. However, sometimes it can be quite hard to decide which density best explains the data. We develop a hypothesis test to determine which parametric density can be chosen. This test is based on the Kullback-Leibler divergence between a nonparametric kernel density estimator and the parametric densities. The distribution of the test statistic will be determined by means of bootstrap with variance stabilization.

### 8.3.1 Introduction to the Problem

One of the key issues in the evolution of MOSFET technology is device scaling (Hu, 1993). Thin oxide layers are an important part of most microelectronic devices. They serve as insulating gate material in transistors. When these very thin oxide layers are subjected to voltages of a few volts, due to the induced electrical field, a gradual degradation of the oxide properties occurs and finally device failure is followed.

Different phases before the final device failure are shown in Figure 8.8. Due to the stress, defects are generated in the oxide which can create conduction paths with multiple traps where current flows through. This phenomenon is observed as a small current jump and is referred to as soft-breakdown. In the following part, the wear-out phase, more defects are generated along the existing conduction paths resulting in the wear out of the paths. This phenomenon is observed as fluctuation of current until device failure occurs. At this final stage, called hard-breakdown, abrupt increase of the current (up to mA) is observed. The time-to-soft-breakdown ( $t_{\text{SBD}}$ ), wear out time ( $t_{\text{WO}}$ ) and the time-to-hard-breakdown ( $t_{\text{HBD}}$ ) are random variables. Understanding the related distributions of oxide breakdown and its preceding degradation phase are needed in order to predict the lifetime at operating conditions. SBD is well known to be Weibull distributed (Wu et al., 2000). About the distribution of the wear-out phase however, contradictory reports can be found (Kerber et al., 2006). It is even claimed (Wu et al., 2007) that the shape of the wear-out distribution depends on the choice of the failure current criterion.

In this Section, we propose a hypothesis test, based on statistical bootstrap with variance stabilization and a nonparametric kernel density estimator, assisting the researcher to find the best pre-specified parametric distribution of random



**Figure 8.8:** Illustration of the different phases of oxide degradation and breakdown in ultra-thin gate stacks (a.u. denotes arbitrary units).

variables. Further, we illustrate the capability of this technique by means of toy examples.

### 8.3.2 Kernel Density Estimation

In what follows, we describe two possible techniques to estimate a density function  $f$ . First, we state some notations and conventions.

Throughout this Section, it will be assumed that we have a sample  $X_1, \dots, X_n$  of independent, identically distributed observations from a continuous univariate distribution with probability density function  $f$ . Also  $\hat{f}$  will be the kernel estimator with kernel  $K$  and window width (or bandwidth)  $h$ .

#### Parzen-Rosenblatt Kernel Density Estimator

Probably one of the most popular and known methods to estimate a density function is the Parzen-Rosenblatt estimator (Rosenblatt, 1956; Parzen, 1962). This kernel estimator with kernel  $K$  is defined by

$$\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{x - X_i}{h}\right), \tag{8.1}$$

where  $h$  is called the bandwidth or smoothing parameter of the kernel. Also, the kernel  $K$  satisfies the following conditions

$$\int K(u) du = 1, \quad \int uK(u) du = 0, \quad 0 < \int u^2K(u) du < \infty,$$

with  $u = (x - y)/h$ . The estimator (8.1) can be considered as a sum of “bumps” placed above each observation. The resulting estimator is then the sum of all these bumps. The kernel function  $K$  determines the shape of the bumps while the bandwidth  $h$  determines their width.

### Regression View of Density Estimation

Here, we will establish a connection between density estimation and nonparametric regression (Fan and Gijbels, 1996; Wasserman, 2006). This connection can be seen by using a binning technique. Suppose we are interested in estimating the density function  $f$  on an interval  $[a, b]$ . Partition the interval  $[a, b]$  into  $N$  subintervals  $\{I_k, k = 1, \dots, N\}$  of equal length  $\Delta = (b - a)/N$ . Let  $x_k$  be the center of  $I_k$  and  $y_k$  be the proportion of the data  $\{X_i, i = 1, \dots, n\}$  falling in the interval  $I_k$ , divided by the bin length  $\Delta$ . The number of subintervals can be determined by  $N = \lceil (b - a)/3.49 \text{MAD}(X)n^{-1/3} \rceil$ , where  $\lceil \cdot \rceil$  denotes the largest integer. It is clear that the bin counts  $n\Delta y_k$  i.e. the number of sample points falling in the  $k$ th bin, have a binomial distribution (Johnson et al., 1997)

$$n\Delta y_k \sim \text{Bin}(n, p_k) \quad \text{with} \quad p_k = \int_{I_k} f(x)dx, \quad (8.2)$$

with  $p_k$  the probability content of the  $k$ th bin. For a fine enough partition i.e.  $N \rightarrow \infty$ , it can be calculated from (8.2), by using a Taylor series expansion of the density in the  $k$ th bin around the midpoint of the bin, that

$$\mathbf{E}[y_k] \approx f(x_k), \quad \mathbf{Var}[y_k] \approx \frac{f(x_k)}{n\Delta}. \quad (8.3)$$

Consequently, we could regard the density estimation problem as a heteroscedastic nonparametric regression problem based on the data  $\{(x_k, y_k) : k = 1, \dots, N\}$  which are approximately independent (Fan, 1996). The nonparametric regression problem is defined as follows

$$y_k = m(x_k) + \varepsilon_k, \quad \varepsilon_k = e_k \eta(m(x_k), x_k),$$

where  $e_k$  are independent and identically distributed. The function  $\eta$  expresses the heteroscedasticity and  $m$  is an unknown real-valued smooth function that we want to estimate. Often in practice homoscedastic data are preferred. The homoscedasticity can be accomplished via Anscombe’s variance stabilizing transformation (Anscombe, 1948) to the bin count  $c_k = n\Delta y_k$ , i.e.

$$y_k^* = 2\sqrt{c_k + \frac{3}{8}}.$$

The density estimator is then obtained by applying a nonparametric smoother to the transformed data set  $\{(x_k, y_k^*) : k = 1, \dots, N\}$ , and taking the inverse of



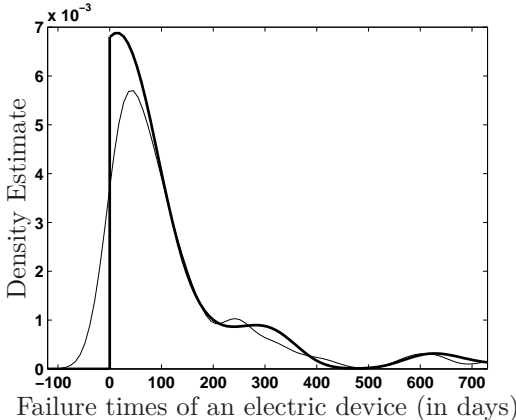
Anscombe’s transformation. Let  $\hat{m}_n^*(x)$  be a regression smoother based on the transformed data. Then the density estimator is defined by

$$\hat{f}(x) = \mathcal{C} \left[ \frac{\hat{m}_n^*(x)^2}{4} - \frac{3}{8} \right]_+, \tag{8.4}$$

where  $\mathcal{C}$  is a normalization constant such that  $\hat{f}(x)$  integrates to 1 and  $[z]_+ = \max(z,0)$ . Then,  $\hat{m}^*$  can be estimated by means of an LS-SVM or any other nonparametric method.

The following example motivates the regression view of density estimation and illustrates an inherent problem of the Parzen-Rosenblatt estimator (8.1).

**Example 8.1** *Given the failure times of an electric device (O’Connor, 1985) measured in days. We want to estimate the failure density by means of the Parzen-Rosenblatt estimator and the regression view of density estimation. Figure 8.9 shows the result of both estimates. It is clear that the Parzen-Rosenblatt estimator (thin line) suffers from a slight drawback when applied to data from long-tailed distributions. Because the bandwidth is fixed across the entire sample, there is a tendency for spurious noise to appear in the tails of the estimates. This example shows this behavior by disregarding the fact that failure times are naturally non-negative. However, this drawback can be overcome by adaptive bandwidths (Silverman, 1996) and/or boundary kernels (Scott, 1992; Wand and Jones, 1995). On the other hand, the LS-SVM based estimate (bold line) for density estimation can deal with this difficulty.*



**Figure 8.9:** Density estimation of failure times of an electric device (measured in days) for the Parzen-Rosenblatt estimator (thin line) and the regression view of density estimation based on LS-SVM (bold line).

**Remark** To determine the number of subintervals  $N$  we have used a reference rule. The literature describes better procedures to find the number of subintervals, see e.g. Park and Marron (1990) for an overview, Wand (1997) for a plug-in type and Devroye and Lugosi (2004) for  $L_1$  optimal bin width selection.

**Remark** Unfortunately, we lack theoretical evidence for the proposed method. However, histogram based density estimators have been frequently studied in literature. Beirlant et al. (1999) considered estimating consistently an unknown probability density function in  $L_1$  from a sample of i.i.d. random variables by means of histogram-based estimators. They showed that their proposed estimator is universally  $L_1$  consistent and attains the kernel estimate rate in the expected  $L_1$  error i.e.  $n^{-2/5}$  in the univariate case. The disadvantage of their density estimate is that it can take on negative values with probability close to 1. In order to overcome this, the idea of Beirlant et al. (1999) was extended to nonnegative piecewise linear histograms (Berlinet and Vajda, 2001) and to generalized piecewise linear histograms (Berlinet et al., 2002).

### 8.3.3 Formulation and Construction of the Hypothesis Test

Given the observations  $X_1, \dots, X_n$ , we are interested in the probability density function  $f$ . One way to proceed is to use nonparametric techniques to estimate  $f$ . On the other hand, in some domains one is particularly interested in acquiring a parametric probability density function (Sahhaf et al., 2009). Often, one has some presumption about the form of the parametric density. For example in the field of reliability analysis of MOSFETs, Weibull, lognormal, exponential, gamma and Gumbel densities are widely used (O'Connor, 1985). As an example, we can formulate the following problem statement: "Which of the given parametric probability density function, Weibull or lognormal, best fits our data?"

To answer this question we propose a methodology based on the comparison between a nonparametric density estimate, obtained by LS-SVM, and a parametric density (chosen by the user). Let us assume that the nonparametric density estimate is our best possible estimator of the unknown true underlying density. Simply put, we can then calculate some kind of distance measure between the nonparametric density estimate and the chosen parametric density. The parametric density corresponding to the smallest distance best fits the given data. Many types of distances between probability densities exist (Tsybakov, 2009) e.g. Hellinger distance, Total Variation (Devroye and Györfi, 1984), Kullback-Leibler (KL) divergence,... In this paper we use the Kullback-Leibler divergence.

**Definition 8.1** The Kullback-Leibler divergence between two distributions  $P$  and  $Q$  is defined as

$$\text{KL}(P, Q) = \int p(x) \log \frac{p(x)}{q(x)} dx, \quad (8.5)$$

where  $p$  and  $q$  denote the densities of  $P$  and  $Q$ .

Typically  $P$  represents our best possible estimate of the “true” distribution of the data e.g. the LS-SVM estimate (8.4) and  $Q$  is the chosen parametric distribution e.g. Weibull, lognormal, etc.

Let  $\hat{F}$  be the LS-SVM estimate of the distribution and let  $\hat{f}$  denote the corresponding estimated density. Also, denote by  $Q_1$  and  $Q_2$  the parametric distributions and by  $q_1$  and  $q_2$  their corresponding parametric densities. These parametric densities are chosen by the user. Consider a situation in which a random sample  $\mathcal{X} = \{X_1, \dots, X_n\}$  is observed from its unspecified probability distribution function  $G_\theta$ , where  $\theta$ , a characteristic of  $G$ , is unknown. We can formulate a hypothesis test as follows:

$$\begin{aligned} H_0 : \theta = \text{KL}(\hat{F}, Q_1) - \text{KL}(\hat{F}, Q_2) \leq \theta_0 \\ \text{vs.} \\ H_1 : \theta = \text{KL}(\hat{F}, Q_1) - \text{KL}(\hat{F}, Q_2) > \theta_0 \end{aligned} \tag{8.6}$$

where  $\theta_0$  is some known bound,  $H_0$  and  $H_1$  represent the null hypothesis and the alternative hypothesis respectively. In other words, if  $H_0$  is accepted, then the parametric density  $q_1$  is closest to the nonparametric estimate and hence is more suited to model the density of the sample than the parametric density  $q_2$  and vice versa.

Let  $\hat{\theta}$  be an estimator of  $\theta$  and  $\hat{\sigma}^2$  an estimator of the variance  $\sigma^2$  of  $\hat{\theta}$ . Define the “studentized” test statistic

$$T_n = \frac{\hat{\theta} - \theta_0}{\hat{\sigma}}.$$

The inclusion of the scale factor  $\hat{\sigma}$  ensures that  $T_n$  is asymptotically pivotal as  $n \rightarrow \infty$  (Hall and Titterton, 1989; Hall and Wilson, 1991). A statistic  $T_n$  is called pivotal if it possesses a fixed probability distribution independent of  $\theta$  (Cramér, 1999; Lehmann and Romano, 2005). Using a pivotal statistic means that we only need to deal with the appropriate standard distribution rather than the whole family of distributions. Note, however, that pivoting often does not hold (Efron and Tibshirani, 1993) unless a variance stabilizing transformation (Tibshirani, 1988) for the parameter estimate of interest is applied first.

Since the distribution function of  $T_n$  under  $H_0$  is not known or cannot be derived analytically, we can use the bootstrap principle (Efron, 1979; Efron and Tibshirani, 1993; Davison and Hinkley, 2003) to test the hypothesis (8.6). The bootstrap principle is based on bootstrap samples. A bootstrap sample  $\mathcal{X}^* = \{X_1^*, \dots, X_n^*\}$  is an unordered collection of  $n$  sample points drawn randomly from  $\mathcal{X}$  with replacement, so that each  $X_i^*$  has probability  $n^{-1}$  of being equal to any of the  $X_j$ ’s. Algorithm 10 states the bootstrap principle to test the hypothesis  $\theta \leq \theta_0$  using variance stabilization.

Note that the constant  $\theta_0$  has been replaced in Step 2 of Algorithm 10 by the estimate of  $\theta$ , i.e.  $\hat{\theta}$ , derived from  $\mathcal{X}$ . This is crucial if the test is to have good power properties. It is also important in the context of the accuracy level of the test (Hall and Titterington, 1989; Hall and Wilson, 1991). Also because we have used variance stabilization we do not need to divide by a standard error estimate in Step 2 and Step 4 since it is constant (Tibshirani, 1988; Efron and Tibshirani, 1993). Another advantage of variance stabilization is that the number of bootstrap resamples decreases significantly. Indeed, the number of bootstrap resamples in this case is  $B_1 \cdot B_2 + B_3 = 100 \cdot 25 + 1000 = 3500$ . In case we did not use this transformation we would have to calculate for each bootstrap resample an estimate of the standard error. Denote by  $B$  the number of bootstrap resamples to calculate the test statistic and  $B^*$  the number of resamples to calculate an estimate of the standard error, then the total number of resamples is  $B \cdot B^* = 1000 \cdot 25 = 25000$ .

---

**Algorithm 10** Testing the hypothesis  $\theta \leq \theta_0$  using variance stabilization

---

1: **Estimation of the variance stabilizing transformation**

- (a) Generate  $B_1$  bootstrap samples  $\mathcal{X}_i^*$  from  $\mathcal{X}$  and calculate  $\hat{\theta}_i^*, i = 1, \dots, B_1$  for  $B_1 = 100$ .
- (b) Generate  $B_2$  bootstrap samples from  $\mathcal{X}_i^*, i = 1, \dots, B_1$ , and calculate  $\hat{\sigma}_i^{*2}$ , a bootstrap estimate for the variance of  $\hat{\theta}_i^*, i = 1, \dots, B_1$ . Set  $B_2 = 25$ .
- (c) Estimate the variance function  $\psi$  by smoothing the values of  $\hat{\sigma}_i^{*2}$  against  $\hat{\theta}_i^*$ .
- (d) Estimate the variance stabilizing transformation  $\Lambda$  by

$$\Lambda = \int_{\theta} [\psi(u)]^{-1/2} du.$$

- 2: **Calculation of the bootstrap statistic.** Generate  $B_3$  bootstrap samples  $\mathcal{X}_i^*$  from  $\mathcal{X}$  and calculate  $T_{n,i}^* = \Lambda(\hat{\theta}^*) - \Lambda(\hat{\theta}), i = 1, \dots, B_3$  for  $B_3 = 1000$ .
  - 3: **Ranking.** Rank the collection  $T_{n,1}^*, \dots, T_{n,B_3}^*$  into increasing order to obtain  $T_{n,(1)}^* \leq \dots \leq T_{n,(B_3)}^*$ .
  - 4: **Test.** Reject  $H_0$  if  $T_n = \Lambda(\hat{\theta}) - \Lambda(\theta_0) > T_{n,(q)}^*$ ,  $q$  determines the level of significance of the test and is given by  $\alpha = \lfloor (B_3 + 1 - q)(B_3 + 1)^{-1} \rfloor$ , with  $\alpha$  the nominal level of significance (Hall and Titterington, 1989) and  $\lfloor \cdot \rfloor$  denotes the smallest integer.
  - 5: **Back-transformation.** If required, transform  $T_{n,i}^*$  and  $T_n$  back by considering the inverse transformation of  $\Lambda$ , i.e.  $\Lambda^{-1}(\Lambda(\hat{\theta}^*) - \Lambda(\hat{\theta}))$  and  $\Lambda^{-1}(\Lambda(\hat{\theta}) - \Lambda(\theta_0))$  respectively.
-

### 8.3.4 Illustrative Examples

In all simulations  $\hat{F}$  denotes the LS-SVM estimate of the distribution. As kernel function we used the Gaussian kernel and the tuning parameters of the LS-SVM (kernel bandwidth  $h$  and regularization parameter  $\gamma$ ) are each bootstrap resample determined by leave-one-out cross-validation. The nominal level of significance  $\alpha$  is set to 0.05,  $B_1 = 100$ ,  $B_2 = 25$  and  $B_3 = 1000$  in Algorithm 10. For each bootstrap resample the parameter(s) of the parametric densities are estimated via Maximum Likelihood (ML).

#### Toy Example 1

In this toy example we generate a data set  $X_i \sim \text{Weib}(1,2)$  for  $i = 1, \dots, 500$ . Suppose we want to test the following hypothesis

$$\begin{aligned} H_0 : \text{KL}(\hat{F}, \text{Weib}) - \text{KL}(\hat{F}, \text{Log}\mathcal{N}) &\leq 0 \\ &\text{vs.} \\ H_1 : \text{KL}(\hat{F}, \text{Weib}) - \text{KL}(\hat{F}, \text{Log}\mathcal{N}) &> 0. \end{aligned}$$

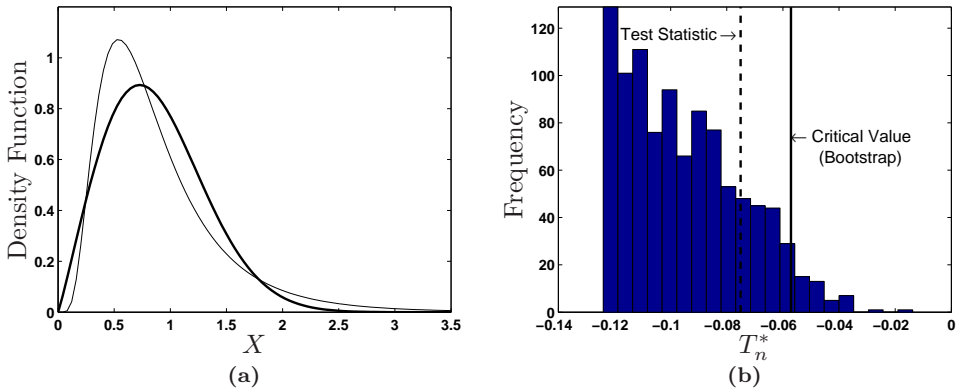
Figure 8.10a shows the lognormal (thin line) and Weibull (bold line) density for the data set with parameters estimated by ML. Figure 8.10b shows the histogram of the test statistic  $T_{n,i}^*, i = 1 \dots, B_3$ . Based on  $T_n < T_{n, (\lfloor (B_3+1)(1-\alpha) \rfloor)}^*$  we cannot reject the null hypothesis  $H_0$  on a significance level  $\alpha = 0.05$ . This indicates that the Weibull distribution is best suited for the given data set, since its KL divergence is the smallest.

#### Toy Example 2

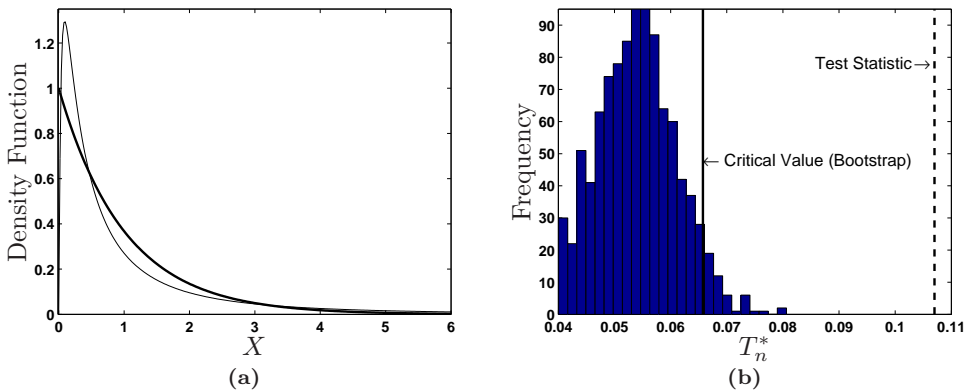
We generate a data set  $X_i \sim \text{Exp}(1)$  for  $i = 1, \dots, 500$ . We test the following hypothesis

$$\begin{aligned} H_0 : \text{KL}(\hat{F}, \text{Log}\mathcal{N}) - \text{KL}(\hat{F}, \text{Exp}) &\leq 0 \\ &\text{vs.} \\ H_1 : \text{KL}(\hat{F}, \text{Log}\mathcal{N}) - \text{KL}(\hat{F}, \text{Exp}) &> 0. \end{aligned}$$

Figure 8.11a shows the lognormal (thin line) and exponential (bold line) density for the data set with parameters estimated by ML. Figure 8.11b shows the histogram of the test statistic  $T_{n,i}^*, i = 1 \dots, B_3$ . Based on  $T_n > T_{n, (\lfloor (B_3+1)(1-\alpha) \rfloor)}^*$  we reject the null hypothesis  $H_0$  on a significance level  $\alpha = 0.05$ . This indicates that the exponential distribution best fits the given data set, since its KL divergence is the smallest.



**Figure 8.10:** (a) lognormal (thin line) and Weibull (bold line) density for the generated data set with parameters estimated by ML; (b) Null distribution of the test statistic. The dashed line indicates the value of  $T_n$  and the solid line indicates the critical value based on the bootstrap samples  $T_n^*$ .



**Figure 8.11:** (a) lognormal (thin line) and exponential (bold line) density for the generated data set with parameters estimated by ML; (b) Null distribution of the test statistic. The dashed line indicates the value of  $T_n$  and the solid line indicates the critical value based on the bootstrap samples  $T_n^*$ .

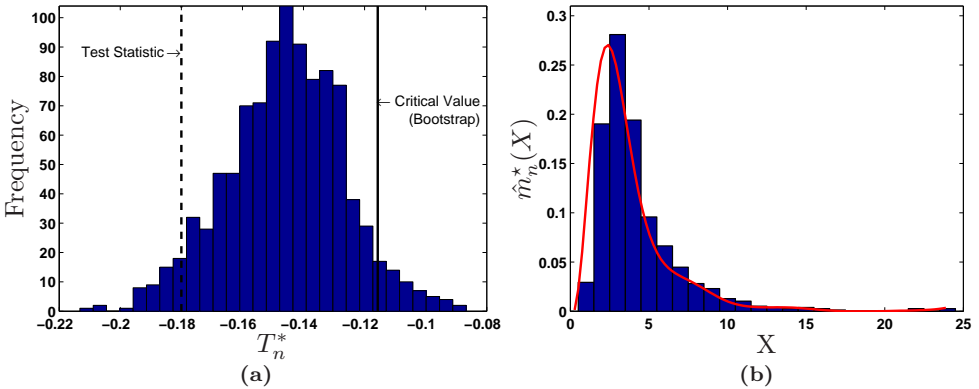
## Real Data Set

Given runaway phase data (Sahhaf et al., 2009) (783 data points). In the field of nano-electronics, there have been many contradictory reports about the distribution of the runaway phase (Kerber et al., 2006; Wu et al., 2007), i.e. Weibull

or lognormal. These distributions are preferred, since the parameters of these distributions can be related to physical mechanisms inside the transistor. We have used the proposed test in order to decide which parametric distribution best fits the given data. We have tested the following hypothesis

$$\begin{aligned}
 H_0 &: \text{KL}(\hat{F}, \text{Log}\mathcal{N}) - \text{KL}(\hat{F}, \text{Weib}) \leq 0 \\
 &\text{vs.} \\
 H_1 &: \text{KL}(\hat{F}, \text{Log}\mathcal{N}) - \text{KL}(\hat{F}, \text{Weib}) > 0.
 \end{aligned}$$

The result is shown in Figure 8.12. Based on this result, we cannot reject the null hypothesis on a significance level of 5% and we can conclude that the runaway phase data is best described by a lognormal distribution.

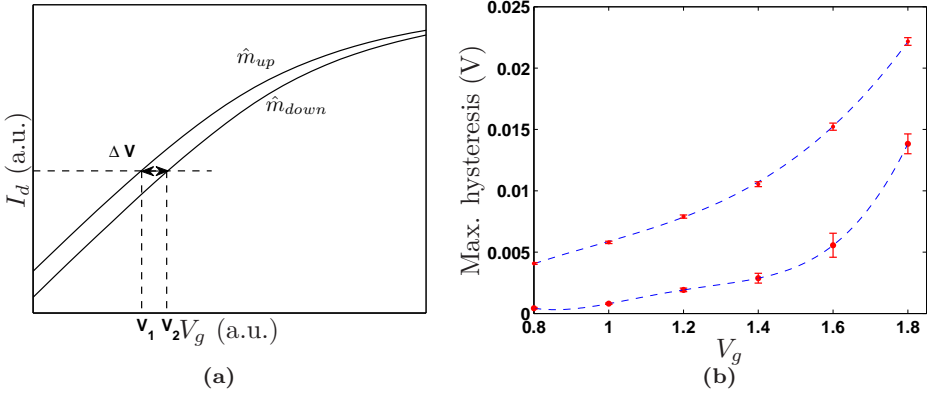


**Figure 8.12:** (a) Null distribution of the test statistic. The dashed line indicates the value of  $T_n$  and the solid line indicates the critical value based on the bootstrap samples  $T_n^*$ ; (b) Density estimate based on LS-SVM (full line).

### 8.4 Finding the Maximum in Hysteresis Curves

In the field of nano-electronics, one often performs so-called hysteresis  $I_d$ - $V_g$  measurements of which an illustration is shown in Figure 8.13a. The maximum extracted  $\Delta V$  from  $I_d$ - $V_g$  hysteresis curves are a measure for the defect density. More specifically, in reliability studies (Sahhaf, 2010), the change of  $\Delta V$  is used in order to study the degradation of devices. As these  $\Delta V$  values can be small, it is important to use an accurate procedure for extracting  $\Delta V$  in order to reduce the analysis-induced errors as much as possible. The idea is to fit both traces (up

and down) by LS-SVM ( $\hat{m}_{up}$  and  $\hat{m}_{down}$ ) and to find the maximum shift in the hysteresis  $\Delta V$ .



**Figure 8.13:** (a) Illustration of the maximum voltage-shift in a hysteresis curve.  $\Delta V$  denotes the maximum shift of the hysteresis curves,  $\hat{m}_{up}$  and  $\hat{m}_{down}$  represent the resulting LS-SVM models of the up and down traces; (b) Extracted maximum hysteresis  $\Delta V$  (dots) with corresponding pointwise standard errors. The dashed curve serves only as a trend line for the maxima.

In order to find voltages  $V_1$  and  $V_2$  (see Figure 8.13a) resulting in the maximum voltage-shift ( $\Delta V = V_2 - V_1$ ) between the up and the down trace in the hysteresis, we use the LS-SVM models describing the up and down traces. Denote by  $\hat{m}_{up}$  and  $\hat{m}_{down}$ , the estimated LS-SVM models for the up and down traces respectively for the given data. The problem of finding the maximum shift  $\Delta V$  is formulated as follows

$$\begin{cases} \max_{V_1, V_2} \mathcal{J}(V_1, V_2) = V_2 - V_1 \\ \text{s.t.} \quad \hat{m}_{up}(V_1) = \hat{m}_{down}(V_2). \end{cases}$$

By using Lagrange multipliers, it is easy to show that the maximum  $\Delta V = V_2 - V_1$  is found by solving the following system

$$\begin{cases} \hat{m}_{up}(V_1) = \hat{m}_{down}(V_2) \\ \hat{m}'_{up}(V_1) = \hat{m}'_{down}(V_2). \end{cases}$$

Figure 8.13b shows the extracted maxima for different hysteresis curves with corresponding pointwise standard error. Based on the extracted  $\Delta V$  as a function of voltage, Sahhaf et al. (2010) specified the position of the defects in the dielectricum of the transistor.



## 8.5 Conclusions

In this Chapter, we illustrated the applicability of LS-SVM in several scientific domains. First, we have shown that FS-LSSVM is a powerful tool for black-box modeling and is capable of handling large data sets. Second, by transforming LS-SVM for regression to a density estimator via a binning technique, we have formulated a hypothesis test based on bootstrap with variance stabilization. This test can assist the researcher to decide which user specified parametric distribution best fits the given data. Finally, we used LS-SVM to determine the maximum shift in hysteresis curves.



## Chapter 9

# Summary, Conclusions and Future Research

### 9.1 Summary and Main Conclusions

#### Summary of Chapter 1

We reviewed the history of parametric and nonparametric regression. Further, we illustrated, by means of some applications, the usefulness of nonparametric regression. Finally, we gave an overview of this thesis and summarized the contributions.

#### Summary of Chapter 2

We reviewed the basic properties of parametric and nonparametric modeling. Several model classes were briefly discussed such as local averaging, local modeling, global modeling and penalized modeling. We have described the assumptions and restrictions on the regression estimates and also we have clarified that any estimate can have an arbitrary slow rate of convergence. Further, we illustrated the curse of dimensionality by means of an example and how it can effect the quality of estimation. Finally, we motivated the basic principle of support vector machines and least squares support vector machines.

### Summary of Chapter 3

We gave an overview of data-driven model selection methods and complexity criteria. Often in practice the chosen criterion will depend on the situation e.g. small or large data set, can we obtain a good noise variance estimation? We also illustrated why these methods should be used in order to acquire suitable tuning parameters with respect to the bias-variance tradeoff. Finally, we made clear that minimizing these criteria is often troublesome since there can be multiple local minima present.

### Main Conclusions/Contributions of Chapter 3

A typical method to estimate the tuning parameters (finding the minimum value of the CV cost function) would define a grid over these parameters of interest and perform CV for each of these grid values. However, three disadvantages come up with the grid-search approach: (i) the limitation of the desirable number of tuning parameters in a model, (ii) practical inefficiency and (iii) discretization fails to take into account the fact that the tuning parameters are continuous. In order to overcome these drawbacks we proposed a two-step optimization approach. First, good initial start values are determined by means of coupled simulated annealing. Second, a fine tuning is performed by means of simplex search. This two-step procedure will result in more optimal tuning parameters and hence better performance.

### Summary of Chapter 4

When considering large scale data sets, we often run into computational problems since the complete kernel matrix cannot be stored in the memory. We investigated if LS-SVM could be solved in the primal space. In order to solve the problem in the primal space, we need a finite approximation of the feature map based on selected prototype vectors. To obtain a suitable selection of these vectors, we maximized the quadratic Rényi entropy. Hence, given such a finite approximation, the problem can be solved as a classical ridge regression problem in the primal space. Finally, the performance of FS-LSSVM is compared to different methods on several data sets. The speed-up achieved by our algorithm is about 10 to 20 times compared to LIBSVM (state-of-the-art library for solving SVMs). We observed that our method requires less prototype vectors than support vectors in SVM, hence resulting in a sparser model.

## Main Conclusions/Contributions of Chapter 4

For LS-SVM, we estimated a finite  $m$ -approximate feature map based on the Nyström approximation so that the problem could be solved in the primal space. In order to select proper prototype vectors, we used the quadratic Rényi entropy. Also, we have illustrated how to select the bandwidth for the entropy estimation in a fast and reliable way using the solve-the-equation plug-in method. Further, we have shown that this entropy criterion with no additional moment constraints is maximized by a uniform density over the input space. In order to select the tuning parameters for large scale data sets, we developed a fast cross-validation procedure. The developed method is able to handle up to one million data points on a current state-of-the-art PC.

## Summary of Chapter 5

We discussed the different approaches used in the literature for achieving robustness in parametric and nonparametric regression models. Further, we illustrated how robustness in the nonparametric case can be obtained by using a least squares cost function via influence functions. Also, we showed, in order to achieve a fully robust procedure, three requirements have to be fulfilled. A robust LS-SVM estimator was obtained via iterative reweighting. We compared four different weight functions and investigated the application in iteratively reweighted LS-SVM. We demonstrated that, by means of simulations and theoretical results, reweighting is useful not only when outliers are present in the data but also to improve stability, especially at heavy tailed distributions. By means of an upper bound for the reduction of the influence function in each step, we revealed the existence of a tradeoff between speed of convergence and the degree of robustness. We demonstrated that the Myriad weight function is highly robust against (extreme) outliers but exhibits a slow speed of convergence. A good compromise between the speed of convergence and robustness can be achieved by using Logistic weights.

## Main Conclusions/Contributions of Chapter 5

In order to achieve a fully robust procedure, we showed that three requirements have to be fulfilled i.e. (i) robust smoother, (ii) bounded kernel and (iii) robust model selection procedure. We compared four different weight functions and investigated the application in iteratively reweighted LS-SVM. We introduced the Myriad reweighting and derived its linear and mode property. By means of an upper bound for the reduction of the influence function in each step, we revealed the existence of a tradeoff between speed of convergence and the degree of robustness. We demonstrated that the Myriad weight function is highly

robust against (extreme) outliers but exhibits a slow speed of convergence. We constructed an empirical maxbias curve of the proposed robust smoother. We showed that its maxbias increases very slightly with the number of outliers and stays bounded right up to the breakdown point which is in strong contrast with the non-robust LS-SVM estimate.

## Summary of Chapter 6

We investigated the possible consequences when the i.i.d. assumption was violated. We illustrated that classical model selection procedures break down in the presence of correlation and not the nonparametric regression method. Since the latter stays consistent when correlation is present in the data, it is not necessary to modify or add extra constraints to the smoother. We proposed a model selection procedure which can handle correlation present in the data without requiring any prior knowledge about its structure. The key to this method are bimodal kernels.

## Main Conclusions/Contributions of Chapter 6

In order to cope with the problem of correlation, we proved that by taking a kernel  $K$  satisfying  $K(0) = 0$ , the correlation structure is successfully removed without requiring any prior knowledge about its structure. Further, we showed both theoretically and experimentally, that by using bimodal kernels the estimate will suffer from increased mean squared error. We developed a class of so-called  $\epsilon$ -optimal class of bimodal kernels, since an optimal bimodal kernel satisfying  $K(0) = 0$  cannot be found, which reduces this effect as much as possible. Finally, we proposed, based on our theoretical justifications, a model selection procedure (CC-CV) for LS-SVM in order to effectively handle correlation in the data.

## Summary of Chapter 7

We discussed the construction of bias-corrected  $100(1 - \alpha)\%$  approximate confidence and prediction intervals (pointwise and simultaneous) for linear smoothers, in particular for LS-SVM. To construct pointwise confidence intervals, we relied on the asymptotic normality of LS-SVM. Further, we discussed a technique called double smoothing to determine the bias without estimating higher order derivatives. In order to obtain uniform or simultaneous confidence intervals we used two techniques i.e Bonferroni/Šidák correction and volume-of-tube formula. We provided extensions of this formula in higher dimensions and discussed how to compute some of the coefficients in practice.

## Main Conclusions/Contributions of Chapter 7

We proved, under certain conditions, the asymptotic normality of LS-SVM. Further, we developed a nonparametric variance estimator which can be related to other well-known nonparametric variance estimators. We illustrated that the width of the bands, based on the volume-of-tube formula, are expanding with increasing dimensionality by means of an example. A Monte Carlo study demonstrated that the proposed bias-corrected  $100(1 - \alpha)\%$  approximate simultaneous confidence intervals achieve the proper empirical coverage rate. Finally, the results for the regression case are extended to the classification case. We illustrated how these intervals can assist the user in assessing the quality of the classifier.

## Summary of Chapter 8

We illustrated the applicability of LS-SVM in several scientific domains. First, we have shown that FS-LSSVM is a powerful tool for black-box modeling and is capable of handling large data sets. Second, by using LS-SVM as a density estimator, we have formulated a hypothesis test based on bootstrap with variance stabilization. This test can assist the researcher to decide which user specified parametric distribution best fits the given data. Finally, we used LS-SVM to determine the maximum shift in hysteresis curves.

## Main Conclusions/Contributions of Chapter 8

We developed a nonparametric density estimation method based on regression and showed that this method is capable of handling densities which difficult to estimate by the classical Parzen method. Further, based on this technique, we developed a hypothesis test based on bootstrap with variance stabilization. We showed the usefulness of the test by means of toy examples and a real life example.

## 9.2 Future Research

The thesis presents some contributions for regression using least-squares support support vector methods. However, still a lot of research has to be done. We will state some future research topics.

- Many consistency properties and rates of convergence are obtained under the i.i.d. assumption. It would be meaningful to investigate under which conditions these results can be extended to the dependent data case.

- In order to construct a finite approximation of the feature map, we selected a subsample of the data has. Although the criterion to select this subsample seems to work good in practice, it cannot determine how large this subsample has to be to obtain good estimates. Therefore, sharper error bounds could be created on the Nyström approximation so that the sample size can be determined beforehand and not in greedy manner. This would greatly simplify the FS-LSSVM algorithm and reduces time when considering large data sets.
- Extending the results obtained in Chapter 6 to random design would be a merit from theoretical point of view.
- Modifying/adapting the volume-of-tube formula to handle correlated and heteroscedastic data in a nonparametric regression framework would widen its application area.
- Extension of our proposed method to obtain confidence intervals to the support vector machine (SVM) case. Our method is based on the linear smoother property of the estimator. However, since SVM is a nonlinear smoother, these results are not valid anymore. Although construction of confidence intervals for SVM can be realized by a (double) bootstrap procedure, it would be beneficial (less time consuming than bootstrap) to work with saddlepoint approximations. Further investigation is needed on how to realize this for nonlinear smoothers.
- Theoretical results could be investigated in case of the LS-SVM for density estimation regarding rate of convergence and consistency properties.



# Appendix A

## Coupled Simulated Annealing

CSA (Xavier-de-Souza et al., 2010) features a new form of acceptance probability functions that can be applied to an ensemble of optimizers. This approach considers several current states which are coupled together by their energies in their acceptance probability function. Also, as distinct from classical SA techniques, parallelism is an inherent characteristic of this class of methods. The objective of creating coupled acceptance probability functions that comprise the energy of many current states or solutions is to generate more information when deciding to accept less favorable solutions.

The following equation describes the acceptance probability function  $A$  with coupling term  $\rho$

$$A_{\theta}(\rho, x_i \rightarrow y_i) = \frac{\exp\left(\frac{-\mathbf{E}(y_i)}{T_k^{ae}}\right)}{\exp\left(\frac{-\mathbf{E}(y_i)}{T_k^{ac}}\right) + \rho},$$

with  $A_{\theta}(\rho, x_i \rightarrow y_i)$  the acceptance probability for every  $x_i \in \Theta$ ,  $y_i \in \Upsilon$  and  $\Upsilon \subset \Theta$ .  $\Upsilon$  denotes the set of all possible states and the set  $\Theta = \{x_i\}_{i=1}^q$  is presented as the set of current states of  $q$  minimizers. The variance  $\sigma^2$  of  $A_{\theta}$  equals

$$\frac{1}{q} \sum_{\forall x_i \in \Theta} A_{\theta}^2 - \frac{1}{q^2}.$$

The coupling term  $\rho$  is given by

$$\rho = \sum_{x_j \in \Theta} \exp\left(\frac{-\mathbf{E}(y_i)}{T_k^{ac}}\right).$$

Hence, CSA considers many current states in the set  $\Theta$ , which is a subset of all possible solutions  $\Upsilon$  and accepts a probing state  $y_i$  based not only on the

corresponding current state  $x_i$  but by considering also the coupling term, which depends on the energy of all other elements of  $\Upsilon$ . Algorithm 11 summarizes the complete CSA procedure:

---

**Algorithm 11** CSA with variance control (Xavier-de-Souza et al., 2010)

---

```

1: Initialization: assign  $q$  random initial solutions to  $\Theta$ ; assess the costs  $\mathbf{E}(x_i), \forall x_i \in \Theta$  and
   evaluate coupling term  $\rho$ ; set initial temperatures  $T_k = T_0$  and  $T_k^{ac} = T_0^{ac}$ ; set time index
    $k = 0, \sigma_D^2 = 0.99 \left( \frac{q-1}{q^2} \right)$  and  $\alpha = 0.05$ 
2: for  $g = 1$  to  $G$  inner iterations do
3:   Generate a probing solution  $y_{ig}$  for each element of  $\Theta$  according to  $y_{ig} = x_{ig} + \varepsilon_{ig}, \forall x_{ig} \in \Theta$ 
   and  $\varepsilon_i$  is a random variable sampled from a given distribution; assess the costs for all
   probing solutions  $E(y_{ig}), \forall i = 1, \dots, q$ ,
4:   For each  $i \in 1, \dots, q$ 
5:   if  $\mathbf{E}(y_{ig}) \leq \mathbf{E}(x_{ig})$  then
6:     accept solution  $y_{ig}$  with probability 1
7:   else
8:     accept solution with probability  $A_\theta(\rho, x_{ig} \rightarrow y_{ig})$ 
9:     if  $A_\theta > r$ , with  $r$  sampled from  $\mathcal{U}[0,1]$  then
10:      set  $x_{ig} = y_{ig}$ 
11:     end if
12:   end if
13:   evaluate  $\rho_g$ 
14: end for
15: Adjust acceptance temperature  $T_k^{ac}$ 
16: if  $\sigma^2 < \sigma_D^2$  then
17:    $T_k^{ac} = T_{k-1}^{ac}(1 - \alpha)$ 
18: else
19:    $T_k^{ac} = T_{k-1}^{ac}(1 + \alpha)$ 
20: end if
21: Decrease generation temperature e.g.  $T_k = \frac{T_0}{k+1}$ 
22: if stopping criterion is met then
23:   Stop
24: else
25:   Go to Step 2
26: end if

```

---

# References

- S. Abe. Sparse least squares support vector training in the reduced empirical feature space. *Pattern Analysis & Applications*, 10(3):203–214, 2007.
- R.K. Adenstedt. On large sample estimation for the mean of a stationary sequence. *The Annals of Statistics*, 2(6):1095–1107, 1974.
- H. Akaike. Statistical predictor identification. *Annals of The Institute of Statistical Mathematics*, 22(1):203–217, 1973.
- D.M. Allen. The relationship between variable selection and data augmentation and a method for prediction. *Technometrics*, 16(1):125–127, 1974.
- N.S. Altman. Kernel smoothing of data with correlated errors. *Journal of the American Statistical Association*, 85(411):749–759, 1990.
- S. An, W. Liu, and S. Venkatesh. Fast cross-validation algorithms for least squares support vector machine and kernel ridge regression. *Pattern Recognition*, 40(8):2154–2162, 2007.
- T.W. Anderson. *The Statistical Analysis of Time Series*. Wiley, New York, 1971.
- F. J. Anscombe. The transformation of Poisson, Binomial and Negative-Binomial data. *Biometrika*, 35:246–254, 1948.
- G.R. Arce. *Nonlinear Signal Processing: A Statistical Approach*. Wiley & Sons, 2005.
- S. Arlot and A. Celisse. A survey of cross-validation procedures for model selection. *Statistics Surveys*, 4:40–79, 2010.
- A. Atkinson and M. Riani. *Robust Diagnostic Regression Analysis*. Springer, 2000.
- C.T.H. Baker. *The Numerical Treatment of Integral Equations*. Oxford Clarendon Press, 1977.
- V. Barnett and T. Lewis. *Outliers in Statistical Data*. Wiley & Sons, 1984.

- G. Baudat and F. Anouar. Kernel based methods and function approximation. *in Proceedings of the International Joint Conference on Neural Networks*, 2: 1244–1249, 2001.
- J. Beirlant, E.J. Dudwicz, L. Györfi, and E.C. van der Meulen. Nonparametric entropy estimation: An overview. *International Journal of the Mathematical Statistics Sciences*, 6:17–39, 1997.
- J. Beirlant, A. Berlinet, and L. Györfi. On piecewise linear density estimators. *Statistica Neerlandica*, 53(3):287–308, 1999.
- R.E. Bellman. *Adaptive Control Processes*. Princeton University Press, 1961.
- K.P. Bennett, J. Hu, J. Xiaoyun, G. Kunapuli, and J.-S. Pang. Model selection via bilevel optimization. *International Joint Conference on Neural Networks*, pages 1922–1929, 2006.
- J. Beran, Y. Feng, and S. Heiler. Modifying the double smoothing bandwidth selector in nonparametric regression. *Statistical Methodology*, 6(5):447–465, 2009.
- R. Beran. Discussion of “jackknife bootstrap and other resampling methods in regression analysis”. *The Annals of Statistics*, 14(4):1295–1298, 1986.
- R.J. Beran. Jackknife approximation to bootstrap estimates. *The Annals of Statistics*, 12(1):101–118, 1984.
- A. Berlinet and I. Vajda. Nonnegative piecewise linear histograms. *Statistics*, 35(4):295–317, 2001.
- A. Berlinet, T. Hobza, and I. Vajda. Generalized piecewise linear histograms. *Statistica Neerlandica*, 56(3):301–313, 2002.
- J.M. Bernardo and A.F.M. Smith. *Bayesian Theory*. Wiley & Sons, 2nd edition, 2000.
- D.P. Bertsekas. *Constrained Optimization and Lagrange Multiplier Methods*. Athena Scientific, 1996.
- R.J. Bhansali, L. Giraitis, and P.S. Kokoska. Convergence of quadratic forms with nonvanishing diagonal. *Statistics & Probability Letters*, 77(7):726–734, 2006.
- P.J. Bickel and M. Rosenblatt. On some global measures of the deviations of density function estimates. *The Annals of Statistics*, 1(6):1071–1095, 1973.
- C.M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

- C.M. Bishop and C.S. Qazaz. Regression with input-dependent noise: a bayesian treatment. In *Advances in Neural Information Processing Systems*, volume 9, pages 347–353. MIT Press, 1997.
- C.L. Blake and C.J. Merz. UCI repository of machine learning databases. <http://archive.ics.uci.edu/ml/datasets.html>, 1998. Irvine, CA.
- S. Bochner. *Lectures on Fourier Integrals*. Princeton University Press, 1959.
- A.W. Bowman. An alternative method of cross-validation for the smoothing of density estimates. *Biometrika*, 71(2):353–360, 1984.
- S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- L. Breiman and J.H. Friedman. Estimating optimal transformations for multiple regression and correlation. *Journal of the American Statistical Association*, 80(391):580–598, 1985.
- L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Wadsworth International Group, 1984.
- M.J. Buckley, G.K. Eagleson, and B.W. Silverman. The estimation of residual variance in nonparametric regression. *Biometrika*, 75(2):189–200, 1988.
- P. Bühlmann and H.R. Künsch. Block length selection in the bootstrap for time series. *Computational Statistics & Data Analysis*, 31(3):295–310, 1999.
- A. Buja, T. Hastie, and R. Tibshirani. Linear smoothers and additive models. *The Annals of Statistics*, 17(2):453–510, 1989.
- P. Burman. A comparative study of ordinary cross-validation,  $v$ -fold crossvalidation and the repeated learning-testing methods. *Biometrika*, 76(3):503–514, 1989.
- P. Burman. Estimation of optimal transformations using  $v$ -fold cross validation and repeated learning-testing methods. *Sankhyā: The Indian Journal of Statistics, Series A*, 52(3):314–345, 1990.
- P. Burman and D. Nolan. A general Akaike-type criterion for model selection in robust regression. *Biometrika*, 82(4):877–886, 1995.
- Y. Cao and Y. Golubev. On oracle inequalities related to smoothing splines. *Mathematical Methods of Statistics*, 15(4):398–414, 2006.
- R.J. Carroll, D. Ruppert, L.A. Stefanski, and C.M. Crainiceanu. *Measurement Error in Nonlinear Models: A Modern Perspective*. Taylor & Francis Group, 2nd edition, 2006.

- G.C. Cawley and N.L.C. Talbot. Reduced rank kernel ridge regression. *Neural Processing Letters*, 16(3):293–302, 2002.
- G.C. Cawley and N.L.C. Talbot. Fast exact leave-one-out cross-validation of sparse least-squares support vector machines. *Neural Networks*, 17(10):1467–1475, 2004.
- C.C. Chang and C.J. Lin. LIBSVM: a library for support vector machines. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>, 2001.
- P. Chaudhuri and J.S. Marron. SIZER for exploration of structures in curves. *Journal of the American Statistical Association*, 94(447):807–823, 1999.
- P. Chaudhuri and J.S. Marron. Scale space view of curve estimation. *The Annals of Statistics*, 28(2):408–428, 2000.
- S.-T. Chiu. Bandwidth selection for kernel estimate with correlated noise. *Statistics & Probability Letters*, 8(4):347–354, 1989.
- A. Christmann and A. Van Messem. Bouligand derivatives and robustness of support vector machines for regression. *Journal of Machine Learning Research*, 9:915–936, 2008.
- A. Christmann and I. Steinwart. Consistency and robustness of kernel-based regression in convex risk minimization. *Bernoulli*, 13(3):799–819, 2007.
- G. Chryssolouris, M. Lee, and A. Ramsey. Confidence interval prediction for neural network models. *IEEE Transactions on Neural Networks*, 7(1):229–232, 1996.
- C.K. Chu and J.S. Marron. Choosing a kernel regression estimator (with discussions). *Statistical Science*, 6(4):404–419, 1991a.
- C.K. Chu and J.S. Marron. Comparison of two bandwidth selectors with dependent errors. *The Annals of Statistics*, 19(4):1906–1918, 1991b.
- B.R. Clark. Uniqueness and Fréchet differentiability of functional solutions to maximum likelihood type equations. *The Annals of Statistics*, 11(4):1196–1205, 1983.
- R.M. Clark. A calibration curve for radiocarbon dates. *Antiquity*, 49:251–266, 1975.
- B. Clarke, E. Fokoué, and H.H. Zhang. *Principles and Theory for Data Mining and Machine Learning*. Springer, 2009.
- W.S. Cleveland and S.J. Devlin. Locally weighted regression: An approach to regression analysis by local fitting. *Journal of the American Statistical Association*, 83(403):596–610, 1988.

- R. Collobert, S. Bengio, and Y. Bengio. A parallel mixture of SVMs for very large scale problems. *Neural Computation*, 14(5):1105–1114, 2002.
- R.D. Cook and S. Weisberg. *Residuals and Influence in Regression*. Chapman & Hall, London, 1982.
- P.-A. Cornillon, N.W. Hengartner, and E. Matzner-Løber. Recursive bias estimation and  $L_2$  boosting. Technical Report LA-UR-09-01177, Los Alamos National Laboratory, <http://arxiv.org/abs/0801.4629>, 2009.
- R. Courant and D. Hilbert. *Methods of Mathematical Physics*. Interscience Publishers, 1953.
- T.M. Cover and J.A. Thomas. *Elements of Information Theory*. John Wiley & Sons, 1991.
- D.R. Cox. Long-range dependence: A review. In *Proceedings of the 50th Anniversary Conference. Statistics: An Appraisal*, pages 55–74. Iowa State University Press, 1984.
- H. Cramér. *Mathematical Methods of Statistics*. Princeton University Press, Princeton, N.J., 1999.
- P. Craven and G. Wahba. Smoothing noisy data with spline functions. *Numerische Mathematik*, 31(4):377–403, 1979.
- C. Croux and G. Haesbroeck. Maxbias curves of robust scale estimators based on subranges. *Metrika*, 53(2):101–122, 2001.
- P.J. Daniell. Observations weighted according to order. *American Journal of Mathematics*, 42(4):222–236, 1920.
- J. Davidson, A. Monticini, and D. Peel. Implementing the wild bootstrap using a two-point distribution. *Economic Letters*, 96(3):309–315, 2007.
- R.A. Davis and U. Gather. The identification of multiple outliers. *Journal of the American Statistical Association*, 88(423):782–792, 1993.
- A.C. Davison and D.V. Hinkley. *Bootstrap Methods and their Application*. Cambridge University Press, 2003. (reprinted with corrections).
- J. De Brabanter. *LS-SVM Regression Modelling and its Applications*. PhD thesis, K.U.Leuven, 2004.
- K. De Brabanter, P. Dreesen, P. Karsmakers, K. Pelckmans, J. De Brabanter, J.A.K. Suykens, and B. De Moor. Fixed-Size LS-SVM applied to the Wiener-Hammerstein benchmark. in *Proceedings of the 15th IFAC Symposium on System Identification (SYSID 2009)*, pages 826–831, July 2009.

- K. De Brabanter, J. De Brabanter, J.A.K. Suykens, and B. De Moor. Optimized fixed-size kernel models for large data sets. *Computational Statistics & Data Analysis*, 54(6):1484–1504, 2010a.
- K. De Brabanter, P. Karsmakers, F. Ojeda, C. Alzate, J. De Brabanter, K. Pelckmans, B. De Moor, J. Vandewalle J., and J.A.K. Suykens. LS-SVMLab Toolbox User’s Guide version 1.7. Technical Report 10-146, K.U.Leuven ESAT-SISTA, 2010b.
- M. Debruyne. An outlier map for support vector machine classification. *The Annals of Applied Statistics*, 3(4):1566–1580, 2009.
- M. Debruyne, S. Serneels, and T. Verdonck. Robustified least squares support vector classification. *Journal of Chemometrics*, 23(9):479–486, 2009.
- M. Debruyne, A. Christmann, M. Hubert, and J.A.K. Suykens. Robustness of reweighted least squares kernel based regression. *Journal of Multivariate Analysis*, 101(2):447–463, 2010.
- E. DeVito, L. Rosasco, A. Caponnetto, M. Piana, and A. Verri. Some properties of regularized kernel methods. *Journal of Machine Learning Research*, 5:1363–1390, 2004.
- L. Devroye. The double kernel method in density estimation. *Annales de l’Institut Henri Poincaré (C) Analyse Non Linéaire*, 25:533–580, 1989.
- L. Devroye and L. Györfi. *Nonparametric Density Estimation: The  $L_1$  View*. Wiley & Sons, 1984.
- L. Devroye and G. Lugosi. *Combinatorial Methods in Density Estimation*. Springer, 2001.
- L. Devroye and G. Lugosi. Bin width selection in multivariate histograms by the combinatorial method. *Test*, 13(1):129–145, 2004.
- L. Devroye and T.J. Wagner. Distribution-free performance bounds for potential function rules. *IEEE Transaction on Information Theory*, 25(5):601–604, 1979.
- L. Devroye, L. Györfi, A. Krzyżak, and G. Lugosi. On the strong universal consistency of nearest neighbor regression function estimates. *The Annals of Statistics*, 22(3):1371–1385, 1994.
- L. Devroye, L. Györfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer, 1996.
- L. Devroye, D. Schäfer, L. Györfi, and H. Walk. The estimation problem of minimum mean squared error. *Statistics & Decisions*, 21(1):15–28, 2003.



- J. Dippon, P. Fritz, and M. Kohler. A statistical approach to case based reasoning, with application to breast cancer data. *Computational Statistics & Data Analysis*, 40(3):579–602, 2002.
- M.B. Dollinger and R.G. Staudte. Influence functions of iteratively reweighted least squares estimators. *Journal of the American Statistical Association*, 86(415):709–716, 1991.
- N.R. Draper and H. Smith. *Applied Regression Analysis*. Wiley & Sons, 3rd edition, 1998.
- G. Dreyfus. *Neural Networks: Methodology and Applications*. Springer, 2005.
- F.Y. Edgeworth. On observations relating to several quantities. *Hermathena*, 6: 279–285, 1887.
- B. Efron. Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, 7(1):1–26, 1979.
- B. Efron. *The Jackknife, the Bootstrap, and Other Resampling Plans*. SIAM, 1982.
- B. Efron. How biased is the apparent error rate of a prediction rule? *Journal of the American Statistical Association*, 81(394):461–470, 1986.
- B. Efron. The length heuristic for simultaneous hypothesis tests. *Biometrika*, 84(1):143–157, 1997.
- B. Efron and R. J. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall, 1993.
- M. Espinoza, J.A.K. Suykens, and B. De Moor. Fixed-size least squares support vector machines : A large scale application in electrical load forecasting. *Computational Management Science (Special Issue on Support Vector Machines)*, 3(2):113–129, 2006.
- M. Espinoza, J.A.K. Suykens, R. Belmans, and B. De Moor. Electric load forecasting - using kernel based modeling for nonlinear system identification. *IEEE Control Systems Magazine (Special Issue on Applications of System Identification)*, 27(5):43–57, 2007.
- R.L. Eubank. *Nonparametric Regression and Spline Smoothing*. Marcel Dekker Inc, 2nd edition, 1999.
- R.L. Eubank and P.L. Speckman. Confidence bands in nonparametric regression. *Journal of the American Statistical Association*, 88(424):1287–1301, 1993.
- J. Fan. Test of significance based on wavelet thresholding and Neyman’s truncation. *Journal of the American Statistical Association*, 91:674–688, 1996.

- J. Fan and I. Gijbels. Variable bandwidth and local linear regression smoothers. *The Annals of Statistics*, 20(4):2008–2036, 1992.
- J. Fan and I. Gijbels. Censored regression : local linear approximations and their applications. *Journal of the American Statistical Association*, 89(426):560–570, 1994.
- J. Fan and I. Gijbels. Data-driven bandwidth selection in local polynomial fitting: Variable bandwidth and spatial adaptation. *Journal of the Royal Statistical Society: Series B*, 57(2):371–394, 1995.
- J. Fan and I. Gijbels. *Local Polynomial Modelling and Its Applications*. Wiley, 1996.
- J. Fan and Q. Yao. Efficient estimation of conditional variance functions in stochastic regression. *Biometrika*, 85(3):645–660, 1998.
- J. Fan and Q. Yao. *Nonlinear Time Series: Nonparametric and Parametric Methods*. Springer, 2003.
- J. Faraway and J. Sun. Simultaneous confidence bands for linear regression with heteroscedastic errors. *Journal of the American Statistical Association*, 90(431):1094–1098, 1995.
- Y. Feng and S. Heiler. A simple bootstrap bandwidth selector for local polynomial fitting. *Journal of Statistical Computation and Simulation*, 79(12):1425–1439, 2009.
- L.T. Fernholz. *von Mises Calculus for Statistical Functionals*. Lecture Notes in Statistics. Springer, 1983.
- F. Ferraty and P. Vieu. *Nonparametric Functional Data Analysis: Theory and Practice*. Springer, 2006.
- J. Fox. *Regression Diagnostics*. SAGE, 1991.
- M. Francisco-Fernández and J.D. Opsomer. Smoothing parameter selection methods for nonparametric regression with spatially correlated errors. *Canadian Journal of Statistics*, 33(2):279–295, 2004.
- M. Francisco-Fernández, J.D. Opsomer, and J.M. Vilar-Fernández. A plug-in bandwidth selector for local polynomial regression estimator with correlated errors. *Journal of Nonparametric Statistics*, 18(1–2):127–151, 2005.
- J. Friedman. Multivariate adaptive regression splines. *The Annals of Statistics*, 19(1):1–67, 1991.
- J. Friedman and W. Stuetzle. Projection pursuit regression. *Journal of the American Statistical Association*, 76(376):817–823, 1981.

- J. Friedman and J. Tukey. A projection pursuit algorithm for exploratory data analysis. *IEEE Transactions on Computers*, C-23(9):881–890, 1974.
- W.A. Fuller. *Measurement Error Models*. John Wiley & Sons, 1987.
- T. Gasser and H.G. Müller. *Smoothing Techniques for Curve Estimation*, volume 757 of *Lecture Notes in Mathematics*, chapter Kernel Estimation of Regression Functions, pages 23–68. Springer, 1979.
- T. Gasser, L. Sroka, and C. Jenner. Residual variance and residual pattern in nonlinear regression. *Biometrika*, 73(3):625–633, 1986.
- S. Geisser. The predictive sample reuse method with applications. *Journal of the American Statistical Association*, 70(350):320–328, 1975.
- M. Girolami. Orthogonal series density estimation and the kernel eigenvalue problem. *Neural Computation*, 14(3):669–688, 2002.
- B.V. Gnedenko and A.N. Kolmogorov. *Limit Distributions for Sums of Independent Random Variables*. Addison-Wesley Publishing Company, 1968. Translated from Russian.
- P.W. Goldberg, C.K. Williams, and C.M. Bishop. Regression with input-dependent noise: A gaussian process treatment. In *Advances in Neural Information Processing Systems*, volume 10, pages 493–499. MIT Press, 1998.
- G.H. Golub and C.F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, 3rd edition, 1996.
- G.M. Golub, M. Heath, and G. Wahba. Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, 21(2):215–223, 1979.
- A. Gretton, R. Herbrich, A.J. Smola, O. Bousquet, and B. Schölkopf. Kernel methods for measuring independence. *Journal of Machine Learning Research*, 6:2075–2129, 2005.
- L. Györfi, M. Kohler, A. Krzyżak, and H. Walk. *A Distribution-Free Theory of Nonparametric Regression*. Springer, 2002.
- L. Györfi, A. Urbán, and I. Vajda. Kernel-based semi-log-optimal empirical portfolio selection strategies. *International Journal of Theoretical and Applied Finance*, 10(3):505–516, 2007.
- L. Györfi, F. Udina, and H. Walk. Nonparametric nearest neighbor based empirical portfolio selection strategies. *Statistics & Decisions*, 26(2):145–157, 2008.
- P. Hall. On bootstrap confidence intervals in nonparametric regression. *The Annals of Statistics*, 20(2):695–711, 1992.

- P. Hall and I. Van Keilegom. Using difference-based methods for inference in nonparametric regression with time-series errors. *Journal of the Royal Statistical Society: Series B*, 65(2):443–456, 2003.
- P. Hall and J.S. Marron. Estimation of integrated squared density derivatives. *Statistics & Probability Letters*, 6(2):109–115, 1987.
- P. Hall and J.S. Marron. On variance estimation in nonparametric regression. *Biometrika*, 77(2):415–419, 1990.
- P. Hall and J.S. Marron. Local minima in cross-validation functions. *Journal of the Royal Statistical Society, Series B (Methodological)*, 53(1):245–252, 1991.
- P. Hall and D.M. Titterington. On confidence bands in nonparametric density estimation and regression. *Journal of Multivariate Analysis*, 27(1):228–254, 1988.
- P. Hall and D.M. Titterington. The effect of simulation order on level accuracy and power of Monte Carlo tests. *Journal of the Royal Statistical Society: Series B*, 51:459–467, 1989.
- P. Hall and S.R. Wilson. Two guidelines for bootstrap hypothesis testing. *Biometrics*, 47:757–762, 1991.
- P. Hall, J.L. Horowitz, and B.-Y. Jing. On blocking rules for the bootstrap with dependent data. *Biometrika*, 82(3):561–574, 1995a.
- P. Hall, S.N. Lahiri, and J. Polzehl. On bandwidth choice in nonparametric regression with both short- and long-range dependent errors. *The Annals of Statistics*, 23(6):1921–1936, 1995b.
- P. Hall, R.C. Wolf, and Q. Yao. Methods for estimating a conditional distribution function. *Journal of the American Statistical Association*, 94(445):154–163, 1999.
- X. Liu L. Hall and K.W. Bowyer. Comments on “A parallel mixture of SVMs for very large scale problems”. *Neural Computation*, 16(7):1345–1351, 2004.
- P.R. Halmos. *Measure Theory*. Springer, 1974.
- F.R. Hampel. *Contributions to the Theory of Robust Estimation*. PhD thesis, University of California, Berkeley, 1968.
- F.R. Hampel. A general definition of qualitative robustness. *The Annals of Mathematical Statistics*, 42(6):1887–1896, 1971.
- F.R. Hampel. The influence curve and its role in robust estimation. *Journal of the American Statistical Association*, 69(346):383–393, 1974.

- F.R. Hampel, E.M. Ronchetti, P.J. Rousseeuw, and W.A. Stahel. *Robust Statistics: The Approach Based On Influence Functions*. Wiley, New York, 1986.
- E.J. Hannan and B.G. Quinn. The determination of the order of an autoregression. *Journal of the Royal Statistical Society, Series B (Methodological)*, 41(2):190–195, 1979.
- W. Härdle. Resampling for inference from curves. *Proceedings of the 47th Session of International Statistical Institute*, pages 59–69, 1989.
- W. Härdle. *Smoothing Techniques with Implementation in S*. Springer, New-York, 1991.
- W. Härdle. *Applied Nonparametric Regression*. Cambridge University Press, 1999. (reprinted).
- W. Härdle and J.S. Marron. Bootstrap simultaneous error bars for nonparametric regression. *The Annals of Statistics*, 19(2):778–796, 1991.
- W. Härdle and A. Tsybakov. Local polynomial estimators of the volatility function in nonparametric autoregression. *Journal of Econometrics*, 81(1):223–242, 1997.
- W. Härdle, P. Hall, and J.S. Marron. How far are automatically chosen regression smoothing parameters from their optimum? *Journal of the American Statistical Association*, 83(401):86–95, 1988.
- W. Härdle, P. Hall, and J.S. Marron. Regression smoothing parameters that are not far from their optimum. *Journal of the American Statistical Association*, 87(417):227–233, 1992.
- J.D. Hart. Differencing as an approximate de-trending device. *Stochastic Processes and their Applications*, 31(2):251–259, 1989.
- J.D. Hart. Kernel regression estimation with time series errors. *Journal of the Royal Statistical Society: Series B*, 53(1):173–187, 1991.
- J.D. Hart and T.E. Wehrly. Kernel regression estimation using repeated measurements data. *Journal of the American Statistical Association*, 81(396):1080–1088, 1986.
- T. Hastie and R. Tibshirani. *Generalized Additive Models*. Chapman & Hall, 1990.
- T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2nd edition, 2009.
- E. Hermann, T. Gasser, and A. Kneip. Choice of bandwidth for kernel regression when residuals are correlated. *Biometrika*, 79(4):783–795, 1992.

- A.E. Hoerl and R.W. Kennard. Ridge regression: biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.
- R.C. Holte. Very simple classification rules perform well on most commonly used datasets. *Machine Learning*, 11(1):63–90, 1993.
- C. Hu. Future CMOS scaling and reliability. *Proc. of the IEEE*, 81(5):682–689, 1993.
- P.J. Huber. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 35(1):73–101, 1964.
- P.J. Huber. A robust version of the probability ratio test. *The Annals of Mathematical Statistics*, 36(6):1753–1758, 1965.
- P.J. Huber. Robust confidence limits. *Probability Theory And Related Fields*, 10(4):269–278, 1968.
- P.J. Huber and E.M. Ronchetti. *Robust Statistics*. Wiley, 2nd edition, 2009.
- P.J. Huber and V. Strassen. Minimax tests and the Neyman-Pearson lemma for capacities. *The Annals of Statistics*, 1(2):251–263, 1973.
- P.J. Huber and V. Strassen. Minimax tests and the Neyman-Pearson lemma for capacities (correction of proof 4.1). *The Annals of Statistics*, 2(1):223–224, 1974.
- M. Hubert. Multivariate outlier detection and robust covariance matrix estimation - discussion. *Technometrics*, 43(3):303–306, 2001.
- M. Hubert, P.J. Rousseeuw, and K. Vanden Branden. ROBPCA: A new approach to robust principal components analysis. *Technometrics*, 47(1):64–79, 2005.
- C.M. Hurvich and C-L. Tsai. Regression and time series model selection in small samples. *Biometrika*, 76(2):297–307, 1989.
- B. Huyck, K. De Brabanter, F. Logist, J. De Brabanter, J. Van Impe, and B. De Moor. Identification of a pilot scale distillation column: A kernel based approach. Technical Report 10-234, ESAT-SISTA, K.U. Leuven, 2010.
- L. Ingber. Very fast simulated re-annealing. *Journal of Mathematical Computer Modelling*, 12(8):967–973, 1989.
- R. Isermann. *Fault-Diagnosis Systems: An Introduction from Fault Detection to Fault Tolerance*. Springer, 2005.
- L. Jiao, L. Bo, and L. Wang. Fast sparse approximation for least squares support vector machine. *IEEE Transactions on Neural Networks*, 18(3):685–697, 2007.

- T. Joachims. in *B. Schölkopf, C.J.C. Burges and A.J. Smola (Eds.), Advances in Kernel Methods: Support Vector Learning*, chapter Making large-scale SVM practical, pages 169–184. MIT Press, Cambridge, 1999.
- N. Johnson, S. Kotz, and N. Balakrishnan. *Discrete Multivariate Distributions*. Wiley & Sons, 1997.
- G.J. Johnston. Probabilities of maximal deviations for nonparametric regression function estimates. *Journal of Multivariate Analysis*, 12(3):402–414, 1982.
- I.T. Jolliffe. *Principal Component Analysis*. Springer-Verlag, 2nd edition, 2002.
- M.C. Jones and P.J. Foster. Generalized jackknifing and higher order kernels. *Journal of Nonparametric Statistics*, 3(11):81–94, 1993.
- M.C. Jones, J.S. Marron, and S.J. Sheather. A brief survey of bandwidth selection for density estimation. *Journal of the American Statistical Association*, 91(433):401–407, 1996.
- M.I. Jordan and R.A. Jacobs. Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, 6(2):181–214, 1994.
- M.A. Jorgensen. Influence function for iteratively defined statistics. *Biometrika*, 80(2):253–265, 1993.
- J. Jurečková and J. Picek. *Robust Statistical Methods with R*. Chapman & Hall (Taylor & Francis Group), 2006.
- P. Karsmakers, K. Pelckmans, K. De Brabanter, H. Van Hamme, and J.A.K. Suykens. Sparse conjugate directions pursuit with application to fixed-size kernel models. Technical Report 10-63, K.U.Leuven, 2010.
- S.S. Keerthi and S.K. Shevade. SMO algorithm for least-squares SVM formulations. *Neural Computation*, 15(2):487–507, 2003.
- S.S. Keerthi, O. Chapelle, and D. DeCoste. Building support vector machines with reduced classifier complexity. *Journal of Machine Learning Research*, 7:1493–1515, 2006.
- M.G. Kendall. *A Course in the Geometry of  $n$  Dimensions*. Charles Griffin & Company Limited, 1961.
- M.G. Kendall, A. Stuart, and J.K. Ord. *The Advanced Theory of Statistics: Design and Analysis, and Time-Series*, volume 3. Griffin, London, 4th edition, 1983.
- A. Kerber, T. Pople, M. Röhner, K. Mosig, and M. Kerber. Impact of failure criteria on the reliability prediction of CMOS devices with ultrathin gate oxides based on voltage ramp stress. *IEEE Electron Device Letters*, 27(7):609–611, 2006.

- K. Kersting, C. Plagemanna, P. Pfaff, and W. Burgard. Most likely heteroscedastic gaussian process regression. *in Proceedings of the 24th International Conference on Machine Learning (ICML)*, 2007.
- T.Y. Kim, D. Kim, B.U. Park, and D.G. Simpson. Nonparametric detection of correlated errors. *Biometrika*, 91(2):491–496, 2004.
- T.Y. Kim, B.U. Park, M.S. Moon, and C. Kim. Using bimodal kernel inference in nonparametric regression with correlated errors. *Journal of Multivariate Analysis*, 100(7):1487–1497, 2009.
- G. Knafl, J. Sacks, and D. Ylvisaker. Confidence bands for regression functions. *Journal of the American Statistical Association*, 80(391):683–691, 1985.
- M. Kohler. Universal consistency of local polynomial kernel regression estimates. *Annals of The Institute of Statistical Mathematics*, 54(4):879–899, 2002.
- M. Kohler. Nonparametric regression with additional measurement errors in the dependent variable. *Journal of Statistical Planning and Inference*, 136(10):3339–3361, 2006.
- M. Kohler and A. Krzyżak. Nonparametric regression estimates using penalized least squares. *IEEE Transactions on Information Theory*, 47(7):3054–3058, 2001.
- S. Konishi and G. Kitagawa. *Information Criteria and Statistical Modeling*. Springer, 2008.
- T. Krivobokova, T. Kneib, and G. Claeskens. Simultaneous confidence bands for penalized spline estimators. *Journal of the American Statistical Association*, 105(490):852–863, 2010.
- S.R. Kulkarni, S.E. Posner, and S. Sandilya. Data-dependent  $k_n$ -NN and kernel estimators consistent for arbitrary processes. *IEEE Transactions on Information Theory*, 48(10):2785–2788, 2002.
- G. Kunapuli, K.P. Bennett, J. Hu, and J.-S. Pang. *in P.M. Pardalos and P. Hansen (Eds.), Data Mining and Mathematical Programming*, chapter Bilevel model selection for support vector machines, pages 129–158. CRM Proceedings & Lecture Notes Volume 45. American Mathematical Society, 2008.
- H. Künsch. The jackknife and the bootstrap for general stationary observations. *The Annals of Statistics*, 17(3):1217–1261, 1989.
- H. Künsch, J. Beran, and F. Hampel. Contrasts under long-range correlations. *The Annals of Statistics*, 21(2):943–964, 1993.



- J.C. Lagarias, J.A. Reeds, M.H. Wright, and P.E. Wright. Convergence properties of the Nelder-Mead simplex method in low dimensions. *SIAM Journal of Optimization*, 9(1):112–147, 1998.
- S.N. Lahiri. Theoretical comparisons of block bootstrap methods. *The Annals of Statistics*, 27(1):386–404, 1999.
- S.N. Lahiri. *Resampling Methods for Dependent Data*. Springer, 2003.
- S.N. Lahiri, K. Furukawa, and Y.-D. Lee. A nonparametric plug-in rule for selecting optimal block lengths for block bootstrap methods. *Statistical Methodology*, 4(3):292–321, 2007.
- S.C. Larson. The shrinkage of the coefficient of multiple correlation. *Journal of Educational Psychology*, 22(1):45–55, 1931.
- E. L. Lehmann and J .P. Romano. *Testing Statistical Hypothesis*. Springer, New York, 3rd edition, 2005.
- N. Leonenko and O. Seleznev. Statistical inference for the  $\epsilon$ -entropy and the quadratic rényi entropy. *Journal of Multivariate Analysis*, 101(9):1981–1994, 2010.
- N. Leonenko, L. Pronzato, and V. Savani. A class of rényi information estimators for multidimensional densities. *The Annals of Statistics*, 36(5):2153–2182, 2008a.
- N. Leonenko, L. Pronzato, and V. Savani. Estimation of entropies and divergences via nearest neighbors. *Tatra Mt. Math. Publ.*, 39:265–273, 2008b.
- D.H.-Y. Leung. Cross-validation in nonparametric regression with outliers. *The Annals of Statistics*, 33(5):2291–2310, 2005.
- K.-C. Li. Asymptotic optimality for  $C_p$ ,  $C_L$ , cross-validation and generalized cross-validation: discrete index set. *The Annals of Statistics*, 15(3):958–975, 1987.
- Q. Li and J.S. Racine. *Nonparametric Econometrics: Theory and Practice*. Princeton University Press, 2006.
- Y. Li, C. Lin, and W. Zhang. Improved sparse least-squares support vector machine classifiers. *Neurocomputing*, 69(13–15):1655–1658, 2006.
- R.Y. Liu. Bootstrap procedure under some non-i.i.d. models. *The Annals of Statistics*, 16(4):1696–1708, 1988.
- C. Loader. *Local Regression and Likelihood*. Springer, 1999.
- C. Loader and J. Sun. Robustness of tube formula based confidence bands. *Journal of Computational and Graphical Statistics*, 6(2):242–250, 1997.

- J. López, K. De Brabanter, J.R. Dorronsoro, and J.A.K. Suykens. Sparse LS-SVMs with  $L_0$ -norm minimization. *Accepted for publication in Proc. of the 19th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, 2011.
- G. Lugosi and K. Zeger. Nonparametric estimation via empirical risk minimization. *IEEE Transactions on Information Theory*, 41(3):677–687, 1995.
- M.A. Lukas. Strong robust generalized cross-validation for choosing the regularization parameter. *Inverse Problems*, 24(3):034006 (16pp), 2008.
- Y.P. Mack and H.-G. Müller. Convolution type estimators for nonparametric regression estimation. *Statistics & Probability Letters*, 7(3):229–239, 1989.
- C.L. Mallows. Some comments on  $C_p$ . *Technometrics*, 15(4):661–675, 1973.
- E. Mammen. Bootstrap and wild bootstrap for high dimensional linear models. *The Annals of Statistics*, 21(1):255–285, 1993.
- O.L. Mangasarian and D.R. Musicant. Successive overrelaxation for support vector machines. *IEEE Transactions on Neural Networks*, 10(5):1032–1037, 1999.
- R. Maronna, D. Martin, and V. Yohai. *Robust Statistics: Theory and Methods*. Wiley, 2006.
- J.S. Marron. *Exploring the Limits of Bootstrap*, chapter Bootstrap Bandwidth Selection, pages 249–262. Wiley, 1992.
- P. McCullagh and J.A. Nelder. *Generalized linear models*. Chapman & Hall, 1999.
- A. Meister. *Deconvolution Problems in Nonparametric Statistics*. Springer, 2009.
- J. Mercer. Functions of positive and negative type and their connection with the theory of integral equations. *Philosophical Transactions of the Royal Society A*, 209:415–446, 1909.
- C.A. Micchelli, Y. Xu, and H. Zhang. Universal kernels. *Journal of Machine Learning Research*, 7:2651–2667, 2006.
- S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K.-R. Müller. Fisher discriminant analysis with kernels. in *Proceedings of the 1999 IEEE Signal Processing Society Workshop (Neural Networks for Signal Processing IX)*, pages 41–48, 1999.
- R.M. Mnatsakanov, N. Misra, Sh. Li, and E.J. Harner.  $k$   $n$ -nearest neighbor estimators of entropy. *Mathematical Methods Of Statistics*, 17(3):261–277, 2008.
- J. Moody. *Advances in Neural Information Processing Systems 4*, J.E. Moody, S.J. Hanson and R.P. Lippmann (Eds), chapter The Effective Number of Parameters: An Analysis of Generalization and Regularization in Nonlinear Learning Systems, pages 847–854. Morgan Kauffmann Publishers, 1992.

- H. Müller. Empirical bandwidth choice for nonparametric kernel regression by means of pilot estimators. *Statistical Decisions*, 2:193–206, 1985.
- H.-G. Müller and U. Stadtmüller. Estimation of heteroscedasticity in regression analysis. *The Annals of Statistics*, 15(2):610–625, 1987.
- U. Müller, A. Schick, and W. Wefelmeyer. Estimating the error variance in nonparametric regression by a covariate-matched U-statistic. *Statistics*, 37(3):179–188, 2003.
- E.A. Nadaraya. On estimating regression. *Theory of Probability and its Applications*, 9(1):141–142, 1964.
- J.A. Nelder and R. Mead. A simplex method for function minimization. *The Computer Journal*, 7(4):308–313, 1965.
- Y. Nesterov and A. Nemirovskii. Interior-point polynomial algorithms in convex programming. *SIAM*, 13, 1993.
- M.-H. Neumann. Fully data-driven nonparametric variance estimators. *Statistics*, 25(3):189–212, 1994.
- J. Nocedal and S.J. Wright. *Numerical Optimization*. Springer, 2nd edition, 2006.
- D. Nuyens. *Fast Construction of Good Lattice Rules*. PhD thesis, Department of Computer Science, K.U.Leuven, 2007.
- E.J. Nyström. Über die praktische Auflösung von Integralgleichungen mit Anwendungen auf Randwertaufgaben. *Acta Mathematica*, 54:185–204, 1930.
- P. D. T. O’Connor. *Practical Reliability Engineering*. Wiley & Sons, 1985.
- C.-J. Ong, S. Shao, and J. Yang. An improved algorithm for the solution of the regularization path of support vector machine. *IEEE Transactions on Neural Networks*, 21(3):451–462, 2010.
- J. Opsomer, Y. Wang, and Y. Yang. Nonparametric regression with correlated errors. *Statistical Science*, 16(2):134–153, 2001.
- E. Osuna, R. Freund, and F. Girosi. An improved training algorithm for support vector machines. in *Proceedings of the 1997 IEEE workshop Neural Networks for Signal Processing VII*, pages 276–285, 1997.
- L. Paninski and M. Yajima. Undersmoothed kernel entropy estimators. *IEEE Transactions on Information Theory*, 54(9):4384–4388, 2008.
- G. Papadopoulos, P.J. Edwards, and A.F. Murray. Confidence estimation methods for neural networks: A practical comparison. *IEEE Transactions on Neural Networks*, 12(6):1278–1287, 2001.

- B.U. Park and J.S. Marron. Comparison of data-driven bandwidth selectors. *Journal of the American Statistical Association*, 85(409):66–72, 1990.
- B.U. Park and J.S. Marron. On the use of pilot estimators in bandwidth selection. *Journal of Nonparametric Statistics*, 1(3):231–240, 1992.
- B.U. Park, Y.K. Lee, T.Y. Kim, and C. Park. A simple estimator of error correlation in non-parametric regression models. *Scandinavian Journal of Statistics*, 33(3):451–462, 2006.
- E. Parzen. On estimation of a probability density function and mode. *The Annals of Mathematical Statistics*, 33(3):1065–1076, 1962.
- K. Pelckmans, J. De Brabanter, J.A.K. Suykens, and B. De Moor. The differogram: Non-parametric noise variance estimation and its use for model selection. *Neurocomputing*, 69(1–3):100–122, 2005.
- V.V. Petrov. *Limit Theorems of Probability Theory: Sequences of Independent Random Variables*. Oxford university Press, 2004. Reprinted.
- R.R. Picard and R.D. Cook. Cross-validation of regression models. *Journal of the American Statistical Association*, 79(387):575–583, 1984.
- J. Platt. *Advances in Kernel Methods - Support Vector Learning*, chapter Flat training of support vector machines using sequential minimal optimization, pages 185–208. MIT Press, 1999.
- W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 1993.
- M.B. Priestley and M.T. Chao. Nonparametric function fitting. *Journal of the Royal Statistical Society, Series B*, 34(3):385–392, 1972.
- J.C. Principe, J. Fisher, and D. Xu. in *S. Haykin (Ed.), Unsupervised Adaptive Filtering, volume 1: Blind Source Separation*, chapter Information Theoretic Learning, pages 265–319. John Wiley & Sons, New York, 2000.
- M.H. Quenouille. Notes on bias in estimation. *Biometrika*, 43(3–4):353–360, 1956.
- S. Rajasekaran. On simulated annealing and nested annealing. *Journal of Global Optimization*, 16(1):43–56, 2000.
- B.L.S. Prakasa Rao. *Nonparametric Functional Estimation*. Academic Press, 1983.
- G. Rätsch, T. Onoda, and K.-R. Müller. Soft margin for adaboost. *Machine Learning*, 42(3):287–320, 2001.
- V.C. Raykar and R. Duraiswami. Fast optimal bandwidth selection for kernel density estimation. in *Proceedings of the 2006 SIAM International Conference on Data Mining, Bethesda, Maryland, 2006*.

- V.C. Raykar and R. Duraiswami. in *L. Bottou, O. Chapelle, D. DeCoste and J. Weston (Eds.), Large-Scale Kernel Machines*, chapter The Improved Fast Gauss Transform with Applications to Machine Learning, pages 175–202. MIT Press, 2007.
- C. Reinsch. Smoothing by spline functions. *Numerische Mathematik*, 10:177–183, 1967.
- A. Rényi. On measures of information and entropy. in *Proceedings of the 4th Berkeley Symposium on Mathematics, Statistics and Probability*, pages 547–561, 1961.
- A. Rényi. *Probability Theory*. Dover Publications, 2007.
- J.A. Rice. Bandwidth choice for nonparametric regression. *The Annals of Statistics*, 12(4):1215–1230, 1984.
- J.A. Rice. *Mathematical Statistics and Data Analysis*. Duxbury Press, 2nd edition, 1995.
- S.O. Rice. The distribution of the maxima of a random curve. *American Journal of Mathematics*, 61(2):409–416, 1939.
- J. Rissanen. A universal prior for integers and estimation by minimum description length. *The Annals of Statistics*, 11(2):416–431, 1983.
- C. Ritz and J.C. Streibig. *Nonlinear Regression with R*. Springer, 2008.
- I. Rivals and L. Personnaz. Construction of confidence intervals for neural networks based on least squares estimation. *Neural Networks*, 13(4–5):463–484, 2000.
- E. Ronchetti. Robust model selection in regression. *Statistics & Probability Letters*, 3(1):21–23, 1985.
- M. Rosenblatt. Remarks on some nonparametric estimates of a density function. *The Annals of Mathematical Statistics*, 27(3):832–837, 1956.
- P.J. Rousseeuw and C. Croux. Alternatives to the median absolute deviation. *Journal of the American Statistical Association*, 88(424):1273–1283, 1993.
- P.J. Rousseeuw and A.M. Leroy. *Robust Regression and Outlier Detection*. Wiley, 2003.
- M. Rudemo. Empirical choice of histograms and kernel density estimators. *Scandinavian Journal of Statistics*, 9(2):65–78, 1982.
- D. Ruppert and M.P. Wand. Multivariate locally weighted least squares regression. *The Annals of Statistics*, 22(3):1346–1370, 1994.

- D. Ruppert, M.P. Wand, and R.J. Carroll. *Semiparametric Regression*. Cambridge University Press, 2003.
- S. Sahhaf. *Electrical Defects in High-k/Metal Gate MOS Transistors*. PhD thesis, IMEC, 2010.
- S. Sahhaf, R. Degraeve, Ph. J. Roussel, B. Kaczer, T. Kauerauf, and G. Groeseneken. A new TDDDB reliability prediction methodology accounting for multiple SBD and wear out. *IEEE Transactions on Electron Devices*, 56(7):1424–1432, 2009.
- S. Sahhaf, R. Degraeve, M. Cho, K. De Brabanter, Ph.J. Roussel, M.B. Zahid, and G. Groeseneken. Detailed analysis of charge pumping and  $I_dV_g$  hysteresis for profiling traps in SiO<sub>2</sub>/HfSiO(N). *Microelectronic Engineering*, 87(12):2614–2619, 2010.
- C.J. Saunders, M. Stitson, J. Weston, L. Bottou, B. Schölkopf, and A.J. Smola. *Support Vector Machine Reference Manual*. Department of Computer Science, Royal Holloway University of London, 1998.
- I.J. Schoenberg. Spline functions and the problem of graduation. *in Proceedings of the National Academy of Sciences U.S.A.*, 52:947–950, 1964.
- B. Schölkopf, A.J. Smola, R.C. Williamson, and P.L. Bartlett. New support vector algorithms. *Neural Computation*, 12(5):1207–1245, 2000.
- G. Schwartz. Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464, 1979.
- D.W. Scott. *Multivariate Density Estimation: Theory, Practice, and Visualization*. Wiley, 1992.
- D.W. Scott and G.R. Terrell. Biased and unbiased cross-validation in density estimation. *Journal of the American Statistical Association*, 82(400):1131–1146, 1987.
- D.W. Scott, R.A. Tapia, and J.R. Thompson. Kernel density estimation revisited. *Nonlinear Analysis: Theory, Methods & Applications*, 1(4):339–372, 1977.
- G.A.F. Seber and C.J. Wild. *Nonlinear Regression*. Wiley & Sons, 2003.
- A. Sen and M. Srivastava. *Regression Analysis: Theory, Methods and Applications*. Springer, 1990.
- R.J. Serfling. *Approximation Theorems of Mathematical Statistics*. John Wiley & Sons, 1980.
- C.E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 1948.

- J. Shao. Linear model selection by cross-validation. *Journal of the American Statistical Association*, 88(422):486–494, 1993.
- J. Shao. *Mathematical Statistics*. Springer, 2nd edition, 2003.
- S.J. Sheather and M.C. Jones. A reliable data-based bandwidth selection method for kernel density estimation. *Journal of the Royal Statistical Society B*, 53(3): 683–690, 1991.
- A.N. Shiryaev. *Probability*. Springer, 2nd edition, 1996.
- B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman & Hall, 1996. Reprinted from 1992.
- B.W. Silverman. Choosing the window width when estimating a density. *Biometrika*, 65(1):1–11, 1978.
- B.W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman & Hall, 1986.
- J.S. Simonoff. *Smoothing Methods in Statistics*. Springer, 1996.
- A.J. Smola and B. Schölkopf. Sparse greedy matrix approximation for machine learning. in *Proceedings of the 17th International Conference on Machine Learning*, pages 911–918, 2000.
- U. Stadtmüller and A. Tsybakov. Nonparametric recursive variance estimation. *Statistics*, 27(1–2):55–63, 1995.
- I. Steinwart. On the influence of the kernel on the consistency of support vector machines. *Journal of Machine Learning Research*, 2:67–93, 2001.
- I. Steinwart and A. Christmann. *Support Vector Machines*. Springer, 2008.
- C.J. Stone. Consistent nonparametric regression. *The Annals of Statistics*, 5(4): 595–645, 1977.
- M. Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society, Series B (Methodological)*, 36(2):11–147, 1974.
- J. Sun. Probabilities of the maxima of gaussian random fields. *The Annals of Probability*, 21(1):852–855, 1993.
- J. Sun and C.R. Loader. Simultaneous confidence bands for linear regression and smoothing. *Annals of Statistics*, 22(3):1328–1345, 1994.
- J. Sun, J. Raz, and J.J. Faraway. Confidence bands for growth and response curves. *Statistica Sinica*, 9(3):679–698, 1999.

- J. Sun, C. Loader, and W. McCormick. Confidence bands in generalized linear models. *The Annals of Statistics*, 28(2):429–460, 2000.
- Z. Sun, Z. Zhang, H. Wang, and M. Jiang. Cutting plane method for continuously constrained kernel-based regression. *IEEE Transactions on Neural Networks*, 21(2):238–247, 2010.
- J.A.K. Suykens and J. Vandewalle. Least squares support vector machine classifiers. *Neural Processing Letters*, 9(3):293–300, 1999.
- J.A.K. Suykens, L. Lukas, and J. Vandewalle. Sparse approximation using least squares support vector machines. *International Symposium on Circuits and Systems (IS-CASS'00)*, pages 757–760, 2000.
- J.A.K. Suykens, J. Vandewalle, and B. De Moor. Intelligence and cooperative search by coupled local minimizers. *International Journal of Bifurcation and Chaos*, 11(8):2133–2144, 2001.
- J.A.K. Suykens, J. De Brabanter, L. Lukas, and J. Vandewalle. Weighted least squares support vector machines : Robustness and sparse approximation. *Neurocomputing*, 48(1–4):85–105, 2002.
- J.A.K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, and J. Vandewalle. *Least Squares Support Vector Machines*. World Scientific, Singapore, 2002.
- J.A.K. Suykens, M.E. Yalcin, and J. Vandewalle. Coupled chaotic simulated annealing processes. *IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 582–585, 2003.
- K. Takeuchi. Distribution of information statistics and criteria for adequacy of models. *Journal of Mathematical Sciences*, 153:12–18, 1976.
- R. Tibshirani and R. Tibshirani. A bias correction for the minimum error rate in cross-validation. *Annals of Applied Statistics*, 3(2):822–829, 2009.
- R. J. Tibshirani. Variance stabilization and the bootstrap. *Biometrika*, 75(3):433–444, 1988.
- H. Tong. *Non-linear Time Series: A Dynamical Systems Approach*. Oxford University Press, Oxford, 1990.
- A.B. Tsybakov. *Introduction to Nonparametric Estimation*. Springer, 2009.
- J. Tukey. Bias and confidence in not quite large samples (abstract). *The Annals of Mathematical Statistics*, 29:614, 1958.
- J.W. Tukey. in I. Olkin (Ed.), *Contributions to Probability and Statistics*, chapter A survey of sampling from contaminated distributions, pages 448–485. Stanford University Press, 1960.



- J.W. Tukey. *Exploratory Data Analysis*. Addison–Wesley, 1977.
- J. Valyon and G. Horváth. A sparse least squares support vector machine classifier. in *Proc. of the International Joint Conference on Neural Networks (IJCNN)*, pages 543–548, 2004.
- T. Van Gestel, J.A.K. Suykens, B. Baesens, S. Viaene, J. Vanthienen, G. Dedene, B. De Moor, and J. Vandewalle. Benchmarking least squares support vector machine classifiers. *Machine Learning*, 54(1):5–32, 2004.
- V.N. Vapnik. *Statistical Learning Theory*. John Wiley & Sons, 1999.
- V.N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 2nd edition, 2000.
- V.N. Vapnik and A. Chervonenkis. A note on one class of perceptrons. *Automation and Remote Control*, 25, 1964.
- V.N. Vapnik and A. Lerner. Pattern recognition using generalized portrait method. *Automation and Remote Control*, 24(6):774–780, 1963.
- M. Vejmelka and K. Hlaváčková-Schindler. Mutual information estimation in higher dimensions: A speed-up of a  $k$ -nearest neighbor based estimator. *Adaptive And Natural Computing Algorithms*, 4431:790–797, 2007. Lecture Notes in Computer Science.
- K.G.H. Vollbrecht and M.M. Wolf. Conditional entropies and their relation to entanglement criteria. *Journal of Mathematical Physics*, 43(9):4299–4307, 2002.
- Z. Šidák. Rectangular confidence region for the means of multivariate normal distributions. *Journal of the American Statistical Association*, 62(318):626–633, 1967.
- G. Wahba. *Spline Models for Observational Data*. SIAM, Philadelphia, PA, 1990.
- G. Wahba and S. Wold. A completely automatic French curve: fitting spline functions by cross-validation. *Communications in Statistics-Theory and Methods*, 4(1):1–17, 1975.
- M.P. Wand. Data-based choice of histogram bin width. *The American Statistician*, 51(1):59–64, 1997.
- M.P. Wand and M.C. Jones. *Kernel Smoothing*. Chapman & Hall, 1995.
- L. Wasserman. *All of Nonparametric Statistics*. Springer, 2006.
- G.S. Watson. Smooth regression analysis. *Sankhyā: The Indian Journal of Statistics, Series A*, 26(4):359–372, 1964.

- H. Weyl. On the volume of tubes. *American Journal of Mathematics*, 61(2): 461–472, 1939.
- E. Whittaker. A new method of graduation. *Proceedings of the Edinburgh Mathematical Society*, 41:63–75, 1923.
- C.K.I Williams and M. Seeger. in *T.K. Leen, T.G. Dietterich and V. Tresp (Eds.), Advances in Neural Information Processing Systems 13*, chapter Using the Nyström Method to Speed Up Kernel Machines, pages 682–688. MIT Press, 2001.
- M. Witczak. *Modelling and Estimation Strategies for Fault Diagnosis of Non-Linear Systems: From Analytical to Soft Computing Approaches*. Lecture Notes in Control and Information Sciences. Springer, 2007.
- C.F.J. Wu. Jackknife, bootstrap and other resampling methods in regression analysis. *The Annals of Statistics*, 14(4):1261–1295, 1986.
- E. Y. Wu, J. Aitken, E. Nowak, A. Vayshenkera, P. Varekampa, G. Hueckel, J. McKenna, D. Harmon, L. K. Han, C. Montrose, and R. Dufresne. Voltage-dependent voltage-acceleration of oxide breakdown for ultra-thin oxides. *IEDM Techn. Dig.*, pages 541–544, 2000.
- E.Y. Wu, S. Tous, and J. Suñé. On the progressive breakdown statistical distribution and its voltage acceleration. *IEDM Techn. Dig.*, pages 493–496, 2007.
- S. Xavier-de-Souza, J.A.K. Suykens, J. Vandewalle, and D. Bollé. Coupled simulated annealing. *IEEE Transactions on Systems, Man and Cybernetics, Part B*, 40(2):320–335, 2010.
- C. Yang, R. Duraiswami, N. Gumerov, and L. Davis. Improved fast Gauss transform and efficient kernel density estimation. *IEEE International Conference on Computer Vision*, pages 464–471, 2003.
- Y. Yang. Consistency of cross validation for comparing regression procedures. *The Annals of Statistics*, 35(6):2450–2473, 2007.
- Z. Ying and K.C. Keong. Fast leave-one-out evaluation and improvement on inference for LS-SVM’s. in *Proceedings of the 17th International Conference on Pattern Recognition (ICPR)*, 3:494–497, 2004.
- K. Zhang, I.W. Tsang, and J.T. Kwok. Improved Nyström low-rank approximation and error analysis. in *Proceedings of the 25th international conference on Machine learning (ICML)*, 2008.
- Y. Zhao and J. Sun. Recursive reduced least squares support vector machines. *Pattern Recognition*, 42(5):837–842, 2009.

# Curriculum vitae

Kris De Brabanter was born on February, 21 1981 in Ninove, Belgium. In 2005, he received the M.Sc. degree in Industriële Wetenschappen (industrieel ingenieur) from the Erasmus Hogeschool Brussel in Belgium. In 2007, he received the M.Sc. degree in Electronic Engineering (burgerlijk ingenieur) from the Katholieke Universiteit Leuven in Belgium. Later on, in October 2007, he started his doctoral studies in kernel based methods at ESAT-SISTA (K.U.Leuven, Belgium) on the topic of “Least Squares Support Vector Regression with Applications to Large-Scale Data: A Statistical Approach”.





Arenberg Doctoral School of Science, Engineering & Technology

Faculty Engineering

Department Electrical Engineering

Research group SISTA - SCD

Kasteelpark Arenberg 10, B-3001 Leuven, Belgium