Studies in Classification, Data Analysis, and Knowledge Organization

Titles in the Series

E. Diday, Y. Lechevallier, M. Schader, P. Bertrand, and B. Burtschy (Eds.)
New Approaches in Classification and Data Analysis. 1994 (out of print)

W. Gaul and D. Pfeifer (Eds.)
From Data to Knowledge. 1995

H.-H. Bock and W. Polasek (Eds.)
Data Analysis and Information Systems. 1996

E. Diday, Y. Lechevallier, and O. Opitz (Eds.)
Ordinal and Symbolic Data Analysis. 1996

R. Klar and O. Opitz (Eds.)
Classification and Knowledge Organization. 1997

C. Hayashi, N. Ohsumi, K. Yajima, Y. Tanaka, H.-H. Bock, and Y. Baba (Eds.)
Data Science, Classification, and Related Methods. 1998

I. Balderjahn, R. Mathar, and M. Schader (Eds.)
Classification, Data Analysis, and Data Highways. 1998

A. Rizzi, M. Vichi, and H.-H. Bock (Eds.)
Advances in Data Science and Classification. 1998

M. Vichi and O. Opitz (Eds.)
Classification and Data Analysis. 1999

W. Gaul and H. Locarek-Junge (Eds.)
Classification in the Information Age. 1999

H.-H. Bock and E. Diday (Eds.)
Analysis of Symbolic Data. 2000

H. A. L. Kiers, J.-P. Rasson, P. J. F. Groenen, and M. Schader (Eds.)
Data Analysis, Classification, and Related Methods. 2000

W. Gaul, O. Opitz, and M. Schader (Eds.)
Data Analysis. 2000

R. Decker and W. Gaul (Eds.)
Classification and Information Processing at the Turn of the Millenium. 2000

S. Borra, R. Rocci, M. Vichi, and M. Schader (Eds.)
Advances in Classification and Data Analysis. 2001

W. Gaul and G. Ritter (Eds.)
Classification, Automation, and New Media. 2002

K. Jajuga, A. Sokołowski, and H.-H. Bock (Eds.)
Classification, Clustering and Data Analysis. 2002

M. Schwaiger and O. Opitz (Eds.)
Exploratory Data Analysis in Empirical Research. 2003

M. Schader, W. Gaul, and M. Vichi (Eds.)
Between Data Science and Applied Data Analysis. 2003

H.-H. Bock, M. Chiodi, and A. Mineo (Eds.)
Advances in Multivariate Data Analysis. 2004

D. Banks, L. House, F. R. McMorris, P. Arabie, and W. Gaul (Eds.)
Classification, Clustering, and Data Mining Applications. 2004

D. Baier and K.-D. Wernecke (Eds.)
Innovations in Classification, Data Science, and Information Systems. 2005

M. Vichi, P. Monari, S. Mignani and A. Montanari (Eds.)
New Developments in Classification and Data Analysis. 2005

D. Baier, R. Decker, and L. Schmidt-Thieme (Eds.)
Data Analysis and Decision Support. 2005

C. Weihs and W. Gaul (Eds.)
Classification – the Ubiquitous Challenge. 2005

Myra Spiliopoulou · Rudolf Kruse
Christian Borgelt · Andreas Nürnberger
Wolfgang Gaul
Editors

From Data and Information Analysis to Knowledge Engineering

Proceedings of the 29th Annual Conference
of the Gesellschaft für Klassifikation e.V.
University of Magdeburg, March 9–11, 2005

With 239 Figures and 120 Tables

Springer

Professor Dr. Myra Spiliopoulou
Otto-von-Guericke-Universität
Magdeburg
Institut für Technische und
Betriebliche Informationssysteme
Universitätsplatz 2
39106 Magdeburg
Germany
myra@iti.cs.uni-magdeburg.de

Professor Dr. Rudolf Kruse
Dr. Christian Borgelt
Jun.-Professor Dr. Andreas
Nürnberger
Otto-von-Guericke-Universität
Magdeburg
Institut für Wissens-
und Sprachverarbeitung
Universitätsplatz 2
39106 Magdeburg
Germany
kruse@iws.cs.uni-magdeburg.de
borgelt@iws.cs.uni-magdeburg.de
nuernb@iws.cs.uni-magdeburg.de

Professor Dr. Wolfgang Gaul
Universität Karlsruhe (TH)
Institut für Entscheidungstheorie
und Unternehmensforschung
76128 Karlsruhe
wolfgang.gaul@wiwi.uni-karlsruhe.de

ISSN 1431-8814
ISBN-10 3-540-31313-3 Springer-Verlag Berlin Heidelberg New York
ISBN-13 978-3-540-31313-7 Springer-Verlag Berlin Heidelberg New York

Cataloging-in-Publication Data
Library of Congress Control Number: 2005938846

Springer · Part of Springer Science+Business Media

springeronline.com

Printed in Germany

Softcover-Design: Erich Kirchner, Heidelberg

SPIN 11584247 43/3153 – 5 4 3 2 1 0 – Printed on acid-free paper

Preface

This volume contains revised versions of selected papers presented during the 29th Annual Conference of the German Classification Society (Gesellschaft für Klassifikation, GfKl'2005). The conference was held at the Otto-von-Guericke-University Magdeburg in March 2005. The theme of the GfKl'2005 was "From Data and Information Analysis to Knowledge Engineering" and encompassed 230 presentations in 74 sessions, including 11 plenary and semi-plenary talks. With 324 attendants from 23 countries, the 29th GfKl conference established a new participation record for the conference series. The conference again provided an attractive interdisciplinary forum for discussions and mutual exchange of knowledge. It was organized in cooperation with the Slovenian Artificial Intelligence Society (SLAIS).

The conference was accompanied by several collocated events. In addition to the Librarians Workshop and the traditional meetings of the working groups, a new important event took place for the first time — the Doctoral Workshop for PhD students. Starting at the GfKl'2004, a Data Mining Competition took place for the second time; for the particularly challenging data analysis problem posed this year 40 solutions were submitted.

The papers in this volume were selected in a second reviewing process after the conference. Each of the 131 submitted long versions of conference contributions was reviewed by two reviewers, and 92 were accepted for this volume. In addition to papers in the fundamental areas Classification, Clustering, and Data Analysis, this volume contains many papers on a wide range of topics with a strong relation to Computer Science. Examples are Text Mining (largest track of the conference as well as in this post-conference volume), Web Mining, Fuzzy Data Analysis, IT Security, Adaptivity and Personalization, and Visualization. Application-oriented topics were addressed in several conference talks. In this volume, the corresponding papers are grouped into the clusters: (1) Economics, Marketing, Banking and Finance, (2) Medicine, Bioinformatics, Biostatistics, (3) Music Analysis. The last paper in this volume reports on the solutions of the winning data mining contestants.

The editors of these proceedings would like to thank the members of the program committee, all reviewers for their vigorous and timely reviewing process, and the authors for their contributions. Special thanks go to the area chairs, who have undertaken the coordination of the reviewing process for their individual tracks and worked under a rigorous time schedule.

The success of the GfKl'2005 conference is due to the effort and involvement of many people. We would like to thank foremostly the local organization team of Silke Reifgerste, Marko Brunzel, Dirk Dreschel, Tanja Falkowski, Folker Folkens, Henner Graubitz, Roland Müller and Rene Schult and their student support team for their hard work in the preparation of this conference and for their support during the event itself. Most cordial thanks go to the organizers of the collocated events: Werner Esswein (TU Dresden) for the organization of the Doctoral Workshop, Hans-J. Hermes (TU Chemnitz) and

Bernd Lorenz (FH München) who organized the Librarians Workshop, Christian Klein (SPSS GmbH Software) and Michael Thess (prudsys AG) for their involvement in the organization of the industrial track and to Jens Strackeljan (Otto-von-Guericke-University Magdeburg) as well as Roland Jonscher and Sigurd Prieur (Sparkassen Rating und Risikosysteme GmbH, Berlin) for the coordination of the Data Mining Competition. The awards for the competition were sponsored by the Deutscher Sparkassen- und Giroverband.

Institutional support has been of paramount importance for the success of the GfKl'2005. Our first thanks go to the Faculty of Computer Science and the Otto-von-Guericke-Universität Magdeburg for providing rooms, facilities, support and assistance to the organization of this conference. We are particularly indebted to the University Rector Klaus Erich Pollmann for his support and involvement. We gratefully acknowledge the support of the city of Magdeburg in organizing the city reception event. In addition, we would like to thank DaimlerChrysler AG and our sponsors Deutscher Sparkassen- und Giroverband, Heins+Partner GmbH, prudsys AG, Springer Verlag GmbH and SPSS GmbH Software for their support. Finally, we would like to thank Christiane Beisel and Martina Bihn of Springer-Verlag, Heidelberg, for their support and dedication to the production of this volume.

The German Classification Society entrusted us with the organization of the GfKl'2005. We are grateful for this honor and for all institutional and personal support provided to us in all phases of the GfKl'2005, from the first planning phase until the print of this volume.

Myra Spiliopoulou,
Rudolf Kruse,
Christian Borgelt,
Andreas Nürnberger,
Wolfgang Gaul

Magdeburg and Karlsruhe,
January 2006

Organization

Chairs

Local Chair
Myra Spiliopoulou (Otto-von-Guericke-University Magdeburg, Germany)

Publication Chair
Rudolf Kruse (Otto-von-Guericke-University Magdeburg, Germany)

Publicity Chair
Andreas Nürnberger (Otto-von-Guericke-University Magdeburg, Germany)

Submission and Book Preparation
Christian Borgelt (Otto-von-Guericke-University Magdeburg, Germany)

Program Chair
Wolfgang Gaul (University of Karlsruhe, Germany)

Program Committee

Hans-Hermann Bock (RWTH Aachen, Germany)
Reinhold Decker (University of Bielefeld, Germany)
Bernard Fichet (University of Aix-Marseille II, France)
Wolfgang Gaul (University of Karlsruhe, Germany)
Rudolf Kruse (Otto-von-Guericke-University Magdeburg, Germany)
Hans-Joachim Lenz (Free University of Berlin, Germany)
Dunja Mladenić (J. Stefan Institute, Slovenia)
Otto Opitz (University of Augsburg, Germany)
Myra Spiliopoulou (Otto-von-Guericke-University Magdeburg, Germany)
Maurizio Vichi (University of Roma — "La Sapienza", Italy)
Claus Weihs (University of Dortmund, Germany)
Klaus-Dieter Wernecke (Charité Berlin, Germany)

Program Sections and Area Chairs

Clustering
Hans-Hermann Bock (RWTH Aachen, Germany)

Discrimination
Gunter Ritter (University Passau, Germany)

Multiway Classification and Data Analysis
Sabine Krolak-Schwerdt (Saarland University, Germany)
Henk A.L. Kiers (University of Groningen, Netherlands)

Multimode Clustering and Dimensionality Reduction
Maurizio Vichi (University Roma — "La Sapienza", Italy)

Robust Methods in Multivariate Statistics
Andrea Cerioli (University of Parma, Italy)

Dissimilarities and Clustering Structures
Bernard Fichet (University of Aix-Marseille II, France)

PLS Path Modeling, PLS Regression and Classification
Natale C. Lauro (University "Federico II" of Napoli, Italy)
V. Esposito Vinzi (University "Federico II" of Napoli, Italy)

Ranking, Multi-label Classification, Preferences
Johannes Fürnkranz (Technical Universtity Darmstadt, Germany)
Eyke Hüllermeier (Philipps-University Marburg, Germany)

Computational Advances in Data Analysis
Hans-Joachim Lenz (Free University Berlin, Germany)

Fuzzy Data Analysis
Rudolf Kruse (Otto-von-Guericke-University Magdeburg, Germany)

Visualization
Patrick J.F. Groenen (Erasmus University Rotterdam, Netherlands)

Classification and Analysis in Data Intensive Scenarios
Gunter Saake (Otto-von-Guericke-University Magdeburg, Germany)

Data Mining and Explorative Multivariate Data Analysis
Luigi D'Ambra (University "Federico II" of Napoli, Italy)
Paulo Giudici (University of Pavia, Italy)

Text Mining
Andreas Nürnberger (Otto-von-Guericke-University Magdeburg, Germany)
Dunja Mladenič (Jozef Stefan Institute Ljubljana, Slovenia)

Web Mining
Myra Spiliopoulou (Otto-von-Guericke-University Magdeburg, Germany)

Adaptivity and Personalization
Andreas Geyer-Schulz (University Karlsruhe, Germany)
Lars Schmidt-Thieme (Albert-Ludwigs-University Freiburg, Germany)

User and Data Authentication in IT Security
Jana Dittmann (Otto-von-Guericke-University Magdeburg, Germany)

Banking and Finance
Hermann Locarek-Junge (Technical University Dresden, Germany)

Marketing
Daniel Baier (Brandenburg University of Technology Cottbus, Germany)
Matthias Meyer (Ludwig-Maximilians-University Mnchen, Germany)

Economics
Otto Opitz (University Augsburg, Germany)

Mining in Business Processes
Claus Rautenstrauch (Otto-von-Guericke-University Magdeburg, Germany)

Bioinformatics and Biostatistics
Berthold Lausen
(Friedrich-Alexander University Erlangen-Nuremberg, Germany)

Classification of High-dimensional Biological and Medical Data
Siegfried Kropf (Otto-von-Guericke-University Magdeburg, Germany)
Johannes Bernarding (Otto-von-Guericke-University Magdeburg, Germany)

Classification with Latent Variable Models
Angela Montanari (University Bologna, Italy)

Medical and Health Sciences
Klaus-Dieter Wernecke (Charité Berlin, Germany)

Music Analysis
Claus Weihs (University Dortmund, Germany)

Industrial Applications and Solutions
Myra Spiliopoulou (Otto-von-Guericke-University Magdeburg, Germany)

Additional Reviewers (in alphabetical order)

Mark Ackermans
Sven Apel
Michael Berthold
Eva Ceulemans
Steffen Bickel
Ulf Brefeld
Christian Döring
Daniel Enache
Tanja Falkowski
María Teresa Gallegos
Michael Gertz
Hans Goebl
Gerard Govaert
Peter Grzybek
Larry Hall
Fred A. Hamprecht
Enrico Hauer
Hartmut Hecker
Christian Hennig
Andreas Hilbert
Andreas Hotho
Frank Klawonn
Juergen Kleffe
Meike Klettke
Peter Kuhbier
Andreas Lang
Berthold Lausen
Wolfgang Lehner
Wolfgang May
Iven Van Mechelen
Alexander Mehler
Paola Monari
Fabian Mörchen
Hans-Joachim Mucha
Daniel Müllensiefen
Gerhard Paaß
Marco Riani
Gunter Ritter
Fabrice Rossi
Kai-Uwe Sattler
Eike Schallehn
Ingo Schmitt
Benno Stein
Gerd Stumme
Michiel van Wezel
Adalbert Wilhelm

Contents

Discriminant Analysis

Classification with Latent Variable Models

Multiway Classification and Data Analysis

Ranking, Multi-label Classification, Preferences

PLS Path Modeling, PLS Regression and Classification

Robust Methods in Multivariate Statistics

Fuzzy Data Analysis

Economics and Mining in Business Processes

Banking and Finance

Marketing

Adaptivity and Personalization

User and Data Authentication in IT Security

Music Analysis

Data Mining Competition

Boosting and ℓ^1-Penalty Methods for High-dimensional Data with Some Applications in Genomics

Peter Bühlmann

Seminar für Statistik, ETH Zürich
CH-8092 Zürich, Switzerland

Abstract. We consider Boosting and ℓ^1-penalty (regularization) methods for prediction and model selection (feature selection) and discuss some relations among the approaches. While Boosting has been originally proposed in the machine learning community (Freund and Schapire (1996)), ℓ^1-penalization has been developed in numerical analysis and statistics (Tibshirani (1996)). Both of the methods are attractive for very high-dimensional data: they are computationally feasible and statistically consistent (e.g. Bayes risk consistent) even when the number of covariates (predictor variables) p is much larger than sample size n and if the true underlying function (mechanism) is sparse: e.g. we allow for arbitrary polynomial growth $p = p_n = O(n^\gamma)$ for any $\gamma > 0$. We demonstrate high-dimensional classification, regression and graphical modeling and outline examples from genomic applications.

1 Introduction

We consider methods which are computationally feasible and statistically accurate for very high-dimensional data. Examples of such data include gene expression experiments where a single expression profile yields a vector of measurements whose dimension p is in the range between 5'000 - 25'000. On the other hand, the number of experiments n is typically in the dozens. Thus, we will have to deal with the case $p \gg n$: the number of variables p is much larger than sample size n. We often refer to this situation as "high-dimensional data".

We consider some unsupervised and supervised problems. In the former, the data are realizations of random variables (usually assumed to be i.i.d. or from a stationary process) $X_1, \ldots, X_n$, where $X_i \in \mathbb{R}^p$. In the supervised context, we have additional (univariate) response variables Y_i, yielding the $(X_1, Y_1), \ldots, (X_n, Y_n)$. In the following, the jth component of $x \in \mathbb{R}^p$ will be denoted by $x^{(j)}$. The main goal for supervised settings is function estimation which includes regression and classification. For example, the target of interest is $\mathbb{E}[Y|X = x]$ for regression (with $Y \in \mathbb{R}$) or $\mathbb{P}[Y = y|X = x]$ for classification (with $Y \in \{0, \ldots, C-1\}$ in classification). We will also demonstrate in section 3.3 a new method for graphical modeling in unsuper-

vised problems: here the goal is to exploit associations among the different (random) variables.

Boosting (Freund and Schapire (1996) and ℓ^1-penalization (Tibshirani (1996)) are very useful techniques for high-dimensional data. From a computational perspective, both have complexity $O(p)$ if $p \gg n$, i.e. *linear* in the dimensionality. Moreover, they have reasonable statistical properties if the true underlying signal or structure is sparse.

2 Boosting

Boosting has been proposed by Freund and Schapire (1996) in the machine learning community for binary classification. Since its inception, it has attracted a lot of attention both in the machine learning and statistics literature.

This is in part due to its excellent reputation as a prediction method. The gradient descent view of boosting as articulated in Breiman (1998) and Friedman et al. (2000) provides a basis for the understanding and new variants of boosting. As an implication, boosting is not only a black-box prediction tool but also an estimation method in specified classes of models, allowing for interpretation of specific model-terms.

2.1 AdaBoost: An Ensemble Method

AdaBoost (Freund and Schapire (1996)) is an ensemble algorithm for binary classification with $Y_i \in \{0, 1\}$. It is (still) the most popular boosting algorithm which exhibits an excellent performance in numerous empirical studies. It works by specifying a base classifier ("weak learner") which is repeatedly applied to iteratively re-weighted data, yielding an ensemble of classifiers $\hat{g}^{[1]}(\cdot), \ldots, \hat{g}^{[m]}(\cdot)$, where each $\hat{g}^{[k]}(\cdot) : \mathbb{R}^p \to \{0, 1\}$. That is:

$$\begin{array}{lll} \text{re-weighted data 1} & \overset{\text{base procedure}}{\longrightarrow} & \hat{g}^{[1]}(\cdot) \\ \text{re-weighted data 2} & \overset{\text{base procedure}}{\longrightarrow} & \hat{g}^{[2]}(\cdot) \\ \cdots & & \cdots \\ \text{re-weighted data m} & \overset{\text{base procedure}}{\longrightarrow} & \hat{g}^{[m]}(\cdot) \end{array}$$

A key issue of AdaBoost is the way how it re-weights the original data; once we have re-weighted data, one simply applies the base procedure to it as if it would be the original dataset. Finally, the AdaBoost classifier

$$\hat{C}^{[m]}_{AdaBoost}(\cdot) = \left(\text{sign}\left(\sum_{j=1}^{m} c_j \hat{g}^{[m]}(\cdot)\right) + 1 \right) /2 \qquad (1)$$

is constructed by a weighted majority vote among the ensemble of individual classifiers. A statistically motivated description can be found in Friedman et al. (2000).

Thus, AdaBoost involves three specifications: (1) the base procedure ("weak learner"), (2) the construction of re-weighted data, (3) the size of the ensemble m. Regarding (1), most popular are classification trees; issue (2) is defined by the AdaBoost description (cf. Friedman et al. (2000)); and the value m in (3) is a simple one-dimensional tuning parameter.

2.2 Boosting and Functional Gradient Descent

Breiman (1998) showed that the somewhat mysterious AdaBoost algorithm can be represented as a steepest descent algorithm in function space which we call functional gradient descent (FGD). This great result opened the door to use boosting in other settings than classification.

In the sequel, boosting and functional gradient descent (FGD) are used as a terminology for the same method or algorithm. The goal is to estimate a function

$$f^*(\cdot) = \text{argmin}_{f(\cdot)}\mathbb{E}[\rho(Y, f(X))] \tag{2}$$

where $\rho(\cdot,\cdot)$ is a real-valued loss function which is typically convex with respect to the second argument. The function class which we minimize over is not of interest for the moment and hence notationally omitted.

Examples of loss functions and their minimizers are given in the following table; each case corresponds to a different boosting algorithm, as explained in section 2.2; see also Friedman et al. (2000).

range spaces	$\rho(y,f)$	$f^*(x)$	algorithm
$y \in \mathbb{R}, f \in \mathbb{R}$	$\lvert y-f\rvert^2$	$\mathbb{E}[Y\lvert X=x]$	L_2Boosting
$y \in \{0,1\}, f \in \mathbb{R}$	$-\log_2(1+e^{-2(2y-1)f})$	$\frac{1}{2}\log\left(\frac{p(x)}{1-p(x)}\right)$	LogitBoost
$y \in \{0,1\}, f \in \mathbb{R}$	$\rho(y,f)=\exp(-(2y-1)f)$	$\frac{1}{2}\log\left(\frac{p(x)}{1-p(x)}\right)$	AdaBoost

For the two last rows, $p(x) = \mathbb{P}[Y = 1|X = x]$.
Boosting pursues some sort of empirical minimization of the empirical risk

$$n^{1-}\sum_{i=1}^{n}\rho(Y_i, f(X_i)) \tag{3}$$

with respect to $f(\cdot)$. To explain this, we introduce next the notion of a base procedure, often called the "weak learner" in the machine learning community.

The Base Procedure Based on some (pseudo-) response variables $\mathbf{U} = U_1, \ldots, U_n$ and predictor variables $\mathbf{X} = X_1, \ldots, X_n$, the base procedure yields a function estimate

$$\hat{g}(\cdot) = \hat{g}_{(\mathbf{U},\mathbf{X})}(\cdot) : \mathbb{R}^p \to \mathbb{R}.$$

Note that we focus here on function estimates with values in $\mathbb{R}$, rather than classifiers with values in $\{0,1\}$ as described in section 2.1. Typically, the function estimate $\hat{g}(x)$ can be thought as an approximation of $\mathbb{E}[U|X = x]$. Most popular base procedures in machine learning are regression trees (or class-probability estimates from classification trees). Among many other alternative choices, the following base procedure is often quite useful in very high-dimensional situations.

Componentwise Linear Least Squares:

$$\hat{g}(x) = \hat{\gamma}_{\hat{\mathcal{S}}} x^{(\hat{\mathcal{S}})},$$
$$\hat{\gamma}_j = \frac{\sum_{i=1}^n U_i X_i^{(j)}}{\sum_{i=1}^n (X_i^{(j)})^2} \ (j = 1, \ldots, p), \quad \hat{\mathcal{S}} = \text{argmin}_{1 \le j \le p} \sum_{i=1}^n (U_i - \hat{\gamma}_j X_i^{(j)})^2.$$

This base procedure fits a linear regression with the one predictor variable which reduces residual sum of squares most.

The Algorithm The generic FGD or boosting algorithm is as follows.

Generic FGD algorithm

Step 1. Initialize $\hat{f}^{[0]}(\cdot) \equiv 0$. Set $m = 0$.

Step 2. Increase m by 1.
Compute negative gradient and evaluate it at $f = \hat{f}^{[m-1]}(X_i)$:

$$U_i = -\frac{\partial}{\partial f} \rho(Y, f)|_{f=\hat{f}^{[m-1]}(X_i)}, \ i = 1, \ldots, n.$$

Step 3. Fit negative gradient vector $U_1, \ldots, U_n$ by using the base procedure, yielding the estimated function

$$\hat{g}^{[m]}(\cdot) = \hat{g}_{\mathbf{U},\mathbf{X}}(\cdot) : \mathbb{R}^p \to \mathbb{R}.$$

The function estimate $\hat{g}^{[m]}(\cdot)$ may be thought of as an approximation of the negative gradient vector $(U_1, \ldots, U_n)$.

Step 4. Do a one-dimensional numerical line-search for the best step-size

$$\hat{s}^{[m]} = \text{argmin}_s \sum_{i=1}^n \rho(Y_i, \hat{f}^{[m-1]}(X_i) + s\hat{g}^{[m]}(X_i)).$$

Step 5. Up-date $\hat{f}^{[m]}(\cdot) = \hat{f}^{[m-1]}(\cdot) + \nu \cdot \hat{s}^{[m]} \hat{g}^{[m]}(\cdot)$ where $0 < \nu \le 1$ is reducing the step-length for following the approximated negative gradient.

Step 6. Iterate Steps 2-5 until $m = m_{stop}$ is reached for some specified stopping iteration m_{stop}.

The factor ν in Step 5 should be chosen "small": our proposal for a default value is $\nu = 0.1$. The FGD algorithm does depend on ν but its choice is not very crucial as long as it is taken to be "small". On the other hand, the stopping iteration m_{stop} is an important tuning parameter of boosting or FGD. Data-driven choices can be done by using cross-validation schemes or internal model selection criteria (Bühlmann (2004)).

By definition, the generic FGD algorithm yields a linear combination of base procedure estimates:

$$\hat{f}^{[m_{stop}]}(\cdot) = \nu \sum_{m=1}^{m_{stop}} \hat{g}^{[m]}(\cdot)$$

which can be interpreted as an estimate from an ensemble scheme, i.e. the final estimator is an average of individual estimates from the base procedure, similar to the formula for AdaBoost in (1). Thus, the boosting solution implies the following constraint for minimizing the empirical risk in (3): the estimate is a linear combination of fits from the base procedure which induces some regularization, see also section 2.6.

2.3 Boosting with the Squared Error Loss: L_2Boosting

When using the squared error loss $\rho(y, f) = |y - f|^2$, the generic FGD algorithm above takes the simple form of refitting the base procedure to residuals of the previous iteration, cf. Friedman (2001).

L_2Boosting

Step 1 (initialization and first estimate). Given data $\{(X_i, Y_i); i = 1, \ldots, n\}$, fit the base procedure

$$\hat{f}^{[1]}(\cdot) = \nu \hat{g}_{(\mathbf{Y},\mathbf{X})}(\cdot).$$

Set $m = 1$.

Step 2. Increase m by 1.
Compute residuals $U_i = Y_i - \hat{f}^{[m-1]}(X_i)$ $(i = 1, \ldots, n)$ and fit the base procedure to the current residuals. The fit is denoted by $\hat{g}^{[m]}(\cdot) = \hat{g}_{(\mathbf{U},\mathbf{X})}(\cdot)$.
Up-date

$$\hat{f}^{[m]}(\cdot) = \hat{f}^{[m-1]}(\cdot) + \nu \hat{g}^{[m]}(\cdot),$$

where $0 < \nu \leq 1$ is a pre-specified step-size parameter. (The line-search, i.e. Step 4 in the generic FGD algorithm from section 2.2, is omitted).

Step 3 (iteration). Repeat Steps 2 and 3 until some stopping value m_{stop} for the number of iterations is reached.

With $m = 2$ (one boosting step) and $\nu = 1$, L_2Boosting has already been proposed by Tukey (1977) under the name "twicing". L_2Boosting with $\nu = 1$ and with the componentwise least squares base procedure for a fixed collection of p basis functions (instead of p predictor variables) coincides with the matching pursuit algorithm of Mallat and Zhang (1993), analyzed also in computational mathematics under the name of "weak greedy algorithm". All these methods are known under the keyword "Gauss-Southwell algorithm". Tukey's (1977) twicing seems to be the first proposal to formulate the Gauss-Southwell idea in the context of a nonparametric smoothing estimator, beyond the framework of linear models (dictionaries of basis functions).

Special emphasis is given here to L_2Boosting with the componentwise linear least squares base procedure: it is a method which does variable/feature selection and employs shrinkage of estimated coefficient to zero (regularization), see also section 2.6.

2.4 A Selective Review of Theoretical Results for Boosting

Asymptotic consistency results for boosting algorithms with early stopping as described in section 2.2 have been given by Jiang (2004) for AdaBoost, Zhang and Yu (2005) for general loss function, and Bühlmann (2004) for L_2Boosting; Bühlmann and Yu (2003) have shown minimax optimality of L_2Boosting in the toy problem of one-dimensional curve estimation. There are quite a few other theoretical analyses of boosting-type methods which use an ℓ^1-penalty instead of early stopping for regularization.

The result in Bühlmann (2004) covers the situation of a very high-dimensional but sparse linear model

$$Y_i = \sum_{j=1}^{p} \beta_j X_i^{(j)} + \varepsilon_i, \ (i = 1, \ldots, n), \tag{4}$$

where $\varepsilon_1, \ldots, \varepsilon_n$ are i.i.d. mean zero variables. High-dimensionality means that the dimension $p = p_n$ is allowed to grow very quickly with sample size n, i.e. $p_n = O(\exp(Cn^{1-\xi}))$ for some $C > 0$ and $0 < \xi < 1$; regarding sparseness, it is required that $\sup_n \sum_{j=1}^{p_n} |\beta_{j,n}| < \infty$ (the coefficients are allowed to change with sample size n, i.e. $\beta_j = \beta_{j,n}$).

2.5 Predictive Performance of Boosting

Most of the first results on the predictive performance on boosting are in classification: they demonstrated that boosting trees is very often substantially better than a single classification tree (cf. Freund and Schapire (1996); Breiman (1998)). In Bühlmann and Yu (2003) it has been pointed out and emphasized that in classical situations, where $p \ll n$ (with p in a reasonable range between 1 and 10), boosting is not better and about as good as

	L_2Boost	FPLR	1-NN	DLDA	SVM
misclassifications	30.50%	35.25%	43.25%	36.12%	36.88%

Table 1. Cross-validated misclassification rates for lymph node breast cancer data. L_2Boosting (L_2Boost), forward variable selection penalized logistic regression (FPLR), 1-nearest-neighbor rule (1-NN), diagonal linear discriminant analysis (DLDA) and a support vector machine (SVM).

more established flexible nonparametric methods. In high-dimensional problems however, boosting performs often much better than more traditional methods.

Binary Classification of Tumor Types based on Gene Expression Data There exists by now a vast variety of proposals for classification based on gene expression data. Boosting is one of the fewer methods which does not require a preliminary dimensionality reduction of the problem (often done in an ad-hoc way selecting the best genes according to a score from a two-sample test, e.g. the best 200 genes). Therefore, boosting can be used as a method for multivariate gene selection (instead of the commonly used principle to quantify the effect of single genes only, e.g. differential expression).

We consider a dataset which monitors $p = 7129$ gene expressions in 49 breast tumor samples using the Affymetrix technology. For each sample, a binary response variable is available, describing the status of lymph node involvement in breast cancer. [1] We use L_2Boosting despite the binary classification structure; a justification for this is given in Bühlmann (2004). We estimate the classification performance by a cross-validation scheme where we randomly divide the 49 samples into balanced training- and test-data of sizes $2n/3$ and $n/3$, respectively, and we repeat this 50 times.

We compare L_2Boosting with the componentwise linear least squares base procedure, step-size $\nu = 0.1$ and some AIC-estimated stopping iteration (see Bühlmann (2004)) with four other classification methods: 1-nearest neighbors, diagonal linear discriminant analysis, support vector machine with radial basis kernel (from the R-package `e1071` and using its default values), and a forward selection penalized logistic regression model (using some reasonable penalty parameter and number of selected genes). For 1-nearest neighbors, diagonal linear discriminant analysis and support vector machine, we preselect the 200 genes which have the best Wilcoxon score in a two-sample problem (estimated from the training dataset only), which is recommended to improve the classification performance. Our L_2Boosting and the forward variable selection penalized regression are run without pre-selection of genes. The results are given in Table 1.

[1] The data are available at `http://data.cgt.duke.edu/west.php`

For this dataset with high misclassification rates (high classification noise), the L_2Boosting is very competitive. Moreover, it is an interesting gene selection method: when applied to the whole dataset and using an AIC-estimated stopping iteration (which equals $m_{stop} = 108$), the method selects 42 out of 7129 genes.

2.6 L_2Boosting and Lasso: Connections and Computational Complexities

In the setting of linear models, Efron et al. (2004) made an intriguing connection between L_2Boosting with componentwise linear least squares and the Lasso (Tibshirani (1996)) defined in formula (5), an ℓ^1-penalized least squares method for linear regression. They consider a version of L_2Boosting, called forward stagewise least squares (denoted in the sequel by FSLR) and they show that for the cases where the design matrix satisfies a "positive cone condition", FSLR with infinitesimally small step-sizes produces a set of solutions which coincides with the set of Lasso solutions when varying the regularization parameter. Furthermore, Efron et al. (2004) proposed the least angle regression (LARS) algorithm as a clever computational short-cut for FSLR and Lasso.

The connection between L_2Boosting and Lasso demonstrates an interesting property of boosting. During the iterations of boosting, we get an "interesting" set of solutions $\{\hat{f}^{[m]}(\cdot); m = 1, 2, \ldots\}$ and corresponding regression coefficients $\{\hat{\beta}^{[m]} \in \mathbb{R}^p;\ m = 1, 2, \ldots\}$. Heuristically, due to the results in Efron et al. (2004), it is "similar" to the set of Lasso solutions $\{\hat{\beta}_\lambda \in \mathbb{R}^p;\ \lambda \in \mathbb{R}^+\}$ when varying the penalty parameter λ, where

$$\hat{\beta}_\lambda = \text{argmin}_{\beta \in \mathbb{R}^p} \sum_{i=1}^{n} (Y_i - \sum_{j=1}^{p} \beta_j X_i^{(j)})^2 + \lambda \sum_{j=1}^{p} |\beta_j|. \tag{5}$$

Computing the set of boosting solutions $\{\hat{f}^{[m]}(\cdot);\ m = 1, 2 \ldots\}$ is computationally quite cheap since every boosting step is typically simple: hence, estimating a good stopping iteration m_{stop} via e.g. cross-validation is computationally attractive, and the computational gain can become even more impressive when using an internal model selection criterion such as AIC (Bühlmann (2004)). Of course, for the special case of linear regression, LARS (Efron et al. (2004)) is computationally even more efficient than boosting.

The computational complexity of boosting in potentially high-dimensional linear models is $O(npm_{stop})$, where m_{stop} denotes the number of iterations in boosting. In the very high-dimensional context with $p \gg n$, a good value for m_{stop} is of negligible order in comparison to the dimension p. Therefore, for computing a good (or optimal) boosting estimator, and if $p \gg n$, the computational complexity is $O(p)$, i.e. linear in the dimensionality p.

The LARS algorithm for computing all Lasso solutions in (5) when varying over the penalty parameter λ has computational complexity

$O(np\min(n,p))$; for $p \gg n$, this becomes $O(p)$ which is again linear in the dimensionality p. We should point out that LARS is quite a bit faster than L_2Boosting with respect to real CPU times.

3 Lasso and ℓ^1-Penalty Methods

We focus here exclusively on linear relationships among (random) variables; this is not restrictive from an L_2-point of view when assuming multivariate normality for the data generating distribution.

3.1 The Lasso for Prediction

We have already defined in (5) the Lasso estimator for the coefficients in a linear model as in (4).

Consistency of the Lasso for a high-dimensional but sparse model, which is similar to the discussion after formula (4) has been given by Greenshtein and Ritov (2004). Together with the computational efficiency for computing all Lasso solutions with the LARS algorithm (see section 2.6), this identifies also the Lasso as a very useful method for high-dimensional linear function estimation and prediction. Some empirical comparisons between the Lasso and L_2Boosting with componentwise linear least squares are presented in Bühlmann (2004).

Binary Classification of Two Tumor Types For the binary classification problem discussed in section 2.5, the cross-validated misclassification error when using the Lasso for a high-dimensional ($p = 7129$) linear model is 27.4% (tuning the penalty parameter via an internal cross-validation) which is slightly better than L_2Boosting and all other methods under consideration. The number of selected genes on the whole dataset is 23, i.e. more sparse than L_2Boosting which selects 42 genes (see also next section 3.2).

3.2 Convex Relaxation with the Lasso and Variable Selection

The Lasso estimator as defined in (5) can also be used for variable/feature selection in a linear model (4), as indicated for the tumor classification example above. Due to the geometry of the ℓ^1-space, with the ℓ^1-norm $\|\beta\|_1 = \sum_j |\beta_j|$, it is well known that the solution of the convex optimization in (5) is sparse:many of the coefficient estimates $\hat{\beta}_j = 0$ if λ is sufficiently large.

Thus, variable selection by checking whether $\hat{\beta}_j$ is zero or not can be easily done. This selection scheme depends on the implementing λ in the optimization in (5). A natural idea would be to choose the λ such that a cross-validation score is minimized. This is, however, not an entirely satisfactory choice as it will select too many variables/features; other choices of λ are described in Meinshausen and Bühlmann (2004).

We should point out that the computational complexity for variable selection with the Lasso is $O(np\min(n,p))$ while the more traditional way of searching over all subset models with a penalized likelihood score (e.g. BIC) requires (in the worst case) to compute 2^p least squares problems. Even when using clever up- and down-dating strategies for optimization of a BIC score, the Lasso computation via the LARS algorithm is much faster involving convex optimization only.

3.3 Gaussian Graphical Modeling with the Lasso

Graphical modeling has become a very useful tool to analyze and display conditional dependencies, i.e. associations, among random variables. We consider the case where the data are i.i.d. realizations from

$$\mathbf{X} = (X^{(1)}, \ldots, X^{(p)}) \sim \mathcal{N}(\mu, \Sigma).$$

A Gaussian graphical model can then be defined as follows. The set of edges consists of the indices $\{1, \ldots, p\}$, corresponding to the components of $\mathbf{X}$. Moreover,

$$\begin{aligned} &\text{there is an undirected edge between node } i \text{ and } j \\ \Leftrightarrow\ &X^{(i)} \text{ conditionally dependent of } X^{(j)} \text{ given all other } \{X^{(k)};\ k \neq i, j\} \\ \Leftrightarrow\ &\Sigma^{-1}_{ij} \neq 0. \end{aligned} \tag{6}$$

The latter equivalence holds because of the Gaussian assumption. Furthermore, the elements from the concentration matrix Σ^{-1} can be linked to regression: $\Sigma^{-1}_{ij}/\Sigma^{-1}_{ii} = \beta_{i;j}$, where

$$X^{(i)} = \beta_{i;j} X^{(j)} + \sum_{k \neq i,j} \beta_{i;k} X^{(k)} + \varepsilon^{(i)}\ (i, j = 1, \ldots, p; i \neq j), \tag{7}$$

where $\varepsilon^{(i)}$ is a mean zero error term. Together with (6), we obtain:

$$\begin{aligned} &\text{there is an undirected edge between node } i \text{ and } j \\ \Leftrightarrow\ &\beta_{i;j} = 0 \text{ or } \beta_{j;i} = 0. \end{aligned}$$

Thus, we can infer the graph from variable selection in regression by doing variable selection in each of the p regression problems in (7). When using a traditional technique such as all subset selection with the BIC score, this would amount to solve (in the worst case) $p2^{p-1}$ least squares problems.

Alternatively, we can use the Lasso which involves convex optimizations only and is orders of magnitudes faster than all subset selection method. In particular, the Lasso method is feasible in very high dimensions with thousands of nodes or variables. For every regression problem as in (7), we compute the estimated coefficients $\hat{\beta}_{i;j}$ (which depend on the choice of λ) and

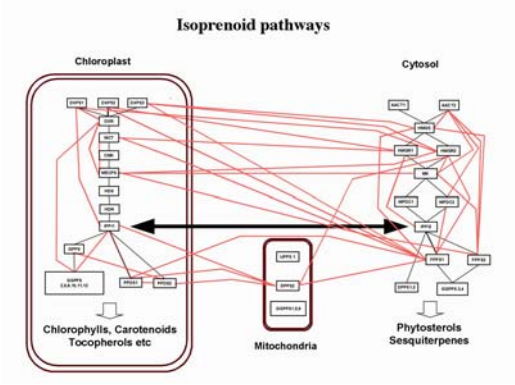

Fig. 1. Estimated graph using the Lasso for the Arabidopsis dataset.

then define a graph estimate as follows:

version 1: there is an undirected edge between node i and j
$\Leftrightarrow \hat{\beta}_{i;j} \neq 0$ or $\hat{\beta}_{j;i} \neq 0$,
version 2: there is an undirected edge between node i and j
$\Leftrightarrow \hat{\beta}_{i;j} \neq 0$ and $\hat{\beta}_{j;i} \neq 0$.

Note the asymmetry in the finite-sample estimates while for the population parameters, it holds that: $\beta_{i;j} = 0 \Leftrightarrow \beta_{j;i} = 0$. Graph estimation with the Lasso depends on the choice of the penalty parameter λ for ℓ^1-penalized regression. The same difficulty arises as in the regression context: the prediction optimal penalty yields too large graphs.

Meinshausen and Bühlmann (2004) prove a consistency result for high-dimensional Gaussian graphical modeling. Roughly speaking, even if the number of variables (nodes) $p = p_n = O(n^\gamma)$ for any $\gamma > 0$, i.e. an arbitrarily fast polynomial growth of the dimension relative to sample size, but assuming that the true graph is sparse, the Lasso graph estimate equals the true graph with probability tending quickly to 1 as sample size n increases.

In Meinshausen and Bühlmann (2004), the Lasso graph estimate has also been compared with forward stepwise selection strategies from the maximum likelihood framework. As a rough summary, the Lasso has better empirical performance (in terms of the ROC curve) if the problem is high-dimensional (relative to sample size n) and the true underlying graph is sparse.

3.4 Estimating a Genetic Network

We applied the Lasso graph estimation method to $n = 118$ gene expression measurements for $p = 39$ genes from two biosynthesis pathways in the model plant Arabidopsis Thaliana.[2] The problem is "fairly high-dimensional" in

[2] The data are available at `http://genomebiology.com/2004/5/11/R92#IDA31O2R`

terms of the ratio n/p. A first goal is to detect potential cross-connections from one to the other pathways.

As seen from Figure 1, the Lasso graph estimator yields quite many edges, i.e. too many for biological interpretations. However, such an estimate can be a first starting point for a more biologically driven analysis, see Wille et al. (2004).

References

BREIMAN, L. (1998): Arcing classifiers. *Ann. Statist., 26, 801–849 (with discussion).*

BÜHLMANN, P. (2004): Boosting for high-dimensional linear models. To appear in the *Ann. Statist.*

BÜHLMANN, P. and YU, B. (2003): Boosting with the L_2loss: regression and classification. *J. Amer. Statist. Assoc., 98, 324–339.*

EFRON, B., HASTIE, T., JOHNSTONE, I. and TIBSHIRANI, R. (2004): Least angle regression. *Ann. Statist., 32, 407–499 (with discussion).*

FREUND, Y. and SCHAPIRE, R.E. (1996): Experiments with a new boosting algorithm. In: *Machine Learning: Proc. Thirteenth International Conference.* Morgan Kauffman, San Francisco, 148–156.

FRIEDMAN, J.H. (2001): Greedy function approximation: a gradient boosting machine. *Ann. Statist., 29, 1189–1232.*

FRIEDMAN, J.H., HASTIE, T. and TIBSHIRANI, R. (2000): Additive logistic regression: a statistical view of boosting. *Ann. Statist., 28, 337–407 (with discussion).*

GREENSHTEIN, E. and RITOV, Y. (2004): Persistency in high dimensional linear predictor-selection and the virtue of over-parametrization. *Bernoulli, 10, 971–988.*

JIANG, W. (2004): Process consistency for AdaBoost. *Ann. Statist., 32, 13–29 (disc. pp. 85–134).*

MALLAT, S and ZHANG, Z. (1993): Matching pursuits with time-frequency dictionaries. *IEEE Trans. Signal Proc., 41, 3397–3415.*

MEINSHAUSEN, N. and BÜHLMANN, P. (2004): High-dimensional graphs and variable selection with the Lasso. To appear in the *Ann. Statist.*

TIBSHIRANI, R. (1996): Regression shrinkage and selection via the lasso. *J. Roy. Statist. Soc., Ser. B, 58, 267–288.*

TUKEY, J.W. (1977): *Exploratory data analysis.* Addison-Wesley, Reading, MA.

WILLE, A., ZIMMERMANN, P., VRANOVÁ, E., FÜRHOLZ, A., LAULE, O., BLEULER, S., HENNIG, L., PRELIĆ, A., VON ROHR, P., THIELE, L., ZITZLER, E., GRUISSEM, W. and BÜHLMANN, P. (2004): Sparse graphical Gaussian modeling of the isoprenoid gene network in Arabidopsis thaliana. *Genome Biology, 5(11) R92, 1-13.*

ZHANG, T. and YU, B. (2005): Boosting with early stopping: convergence and consistency. *Ann. Statist., 33, 1538–1579.*