# Book of abstracts

## ROKS 2013




July 8 - 10, 2013
Leuven, Belgium

European Research Council
Established by the European Commission

KU LEUVEN

FWO Vlaanderen
Fonds Wetenschappelijk Onderzoek
Research Foundation – Flanders

belspo

iMinds
CONNECT.INNOVATE.CREATE

iwt

ROKS 2013

# International Workshop on Advances in Regularization, Optimization, Kernel Methods and Support Vector Machines: theory and applications

July 8-10, 2013

LEUVEN, BELGIUM

## Book of Abstracts




Editors: Johan Suykens, Andreas Argyriou, Kris De Brabanter, Moritz Diehl, Kristiaan Pelckmans, Marco Signoretto, Vanya Van Belle, Joos Vandewalle

Johan Suykens, Andreas Argyriou, Kris De Brabanter, Moritz Diehl, Kristiaan Pelckmans, Marco Signoretto, Vanya Van Belle, Joos Vandewalle (Eds.)
Book of Abstracts of the International Workshop on Advances in Regularization, Optimization, Kernel Methods and Support Vector Machines: theory and applications (ROKS 2013), Leuven, Belgium, July 8-10, 2013

© Katholieke Universiteit Leuven
Department of Electrical Engineering
ESAT-SCD/SISTA
Kasteelpark Arenberg 10
B-3001, Heverlee-Leuven, Belgium

**Layout by:**
Jacqueline De bruyn
Marco Signoretto
Johan Suykens
Vanya Van Belle
Liesbeth Van Meerbeek

ISBN 978-94-6018-700-1
Wettelijk depot D/2013/7515/86

All rights reserved. No part of this book may be reproduced without prior written permission from the publisher.

# Table of Contents

1. Welcome ..... 5
2. Scientific Program ..... 11
3. Invited talk abstracts ..... 23
4. Oral and poster session abstracts ..... 37
5. List of participants ..... 113
6. Practical info ..... 123

# Welcome

# Welcome!

Dear Participants,

Very welcome to the **ROKS 2013** *International Workshop on advances in Regularization, Optimization, Kernel Methods and Support Vector Machines: theory and applications* in Leuven Belgium. In recent years considerable progress has been made in the areas of kernel methods & support vector machines and compressed sensing & sparsity, where convex optimization plays an important common role. The aim of ROKS 2013 is to provide a multi-disciplinary forum where researchers of different communities can meet, to find new synergies along these areas, both at the level of theory and applications. As written in the acronym, the main themes include:

**R**egularization
**O**ptimization
**K**ernel methods
**S**upport vector machines.

We are happy that many distinguished speakers have accepted our invitation to deliver a plenary talk at the occasion of this workshop: *Francis Bach, Stephen Boyd, Martin Jaggi, James Kwok, Yurii Nesterov, Massimiliano Pontil, Justin Romberg, Bernhard Schölkopf, John Shawe-Taylor, Joel Tropp, Ding-Xuan Zhou.* In addition the workshop features five oral sessions organized around the themes: (i) Feature selection and sparsity; (ii) Optimization algorithms; (iii) Kernel methods and support vector machines; (iv) Structured low-rank approximation; (v) Robustness; and two poster sessions together with poster spotlight presentations. The reception in the Salons of the Arenberg castle and the workshop dinner in the Faculty club offer plenty of additional opportunities for interaction and scientific discussions.

The organization of this workshop would have been impossible without the help of many people. We are very grateful to *Jacqueline De bruyn, Liesbeth Van Meerbeek, Elsy Vermoesen, Ida Tassens, Mimi Deprez and several other team members* for the workshop secretariat support, local arrangements and assistance. We thank all Scientific committee members and Reviewers for their valuable comments. It enabled the authors to prepare well-polished extended abstracts. The ROKS 2013 has been organized within the framework of the European Research Council (ERC) Advanced Grant project 290923 A-DATADRIVE-B. We also gratefully acknowledge KU Leuven, FWO, IUAP DYSCO, CoE OPTEC EF/05/006, GOA MANET, iMinds Future Health Department, IWT.

On behalf also of all other organizing committee members *Andreas Argyriou, Kris De Brabanter, Moritz Diehl, Kristiaan Pelckmans, Marco Signoretto, Vanya Van Belle, Joos Vandewalle*, we wish all participants a stimulating workshop with many exciting discussions. Let it roks!

Johan Suykens
Chair ROKS 2013

## Organizing committee

- Johan Suykens, KU Leuven, Chair

- Andreas Argyriou, Ecole Centrale Paris
- Kris De Brabanter, KU Leuven
- Moritz Diehl, KU Leuven
- Kristiaan Pelckmans, Uppsala University
- Marco Signoretto, KU Leuven
- Vanya Van Belle, KU Leuven
- Joos Vandewalle, KU Leuven

## Local arrangements

- Jacqueline De bruyn
- Mimi Deprez
- Ida Tassens
- Elsy Vermoesen
- Liesbeth Van Meerbeek

## Scientific committee

- Carlos Alzate, IBM Research
- Gavin Cawley, University of East Anglia
- Alessandro Chiuso, University of Padova
- Andreas Christmann, University of Bayreuth
- Arnak Dalalyan, ENSAE-CREST
- Francesco Dinuzzo, Max Planck Institute, Tuebingen
- Yonina Eldar, Technion Israel Institute of Technology
- Irene Gijbels, KU Leuven
- Francois Glineur, Catholic University of Louvain
- Kurt Jetter, Universitat Hohenheim
- S. Sathiya Keerthi, Microsoft
- Irwin King, Chinese University of Hong Kong
- Vera Kurkova, Academy of Sciences Czech Republic
- Gert Lanckriet, University of California, San Diego
- Lek-Heng Lim, University of Chicago
- Chih-Jen Lin, National Taiwan University
- Ivan Markovsky, Vrije Universiteit Brussel
- Sayan Mukherjee, Duke University
- Henrik Ohlsson, Linkoping University
- Francesco Orabona, Toyota Technological Institute Chicago
- Pradeep Ravikumar, University of Texas, Austin
- Lorenzo Rosasco, MIT
- Marcello Sanguineti, University of Genova

- Shai Shalev-Shwartz, The Hebrew University of Jerusalem
- Suvrit Sra, Max Planck Institute, Tuebingen
- Ambuj Tewari, University of Michigan
- Sergios Theodoridis, University of Athens
- Ryota Tomioka, The University of Tokyo
- Lieven Vandenberghe, University of California, Los Angeles
- Allessandro Verri, Universita degli Studi di Genova
- Bo Wahlberg, KTH Royal Institute of Technology
- Yuesheng Xu, Syracuse University
- Yiming Ying, University of Exeter
- Ming Yuan, Georgia Institute of Technology
- Luca Zanni, University of Modena and Reggio Emilia

# Scientific Program

# Program

## Monday July 8

12:00-13:00 Registration and welcome coffee in Arenberg castle

13:00-13:10 Welcome by Johan Suykens

13:10-14:00 *Deep-er Kernels*
John Shawe-Taylor (University College London)
[Chair: Kristiaan Pelckmans]

14:00-14:50 *Connections between the Lasso and Support Vector Machines*
Martin Jaggi (Ecole Polytechnique Paris)
[Chair: Andreas Argyriou]

14:50-15:10 Coffee break

15:10-16:40 Oral session 1 : *Feature selection and sparsity*
[Chair: Johan Suykens]

16:40-17:30 *Kernel Mean Embeddings applied to Fourier Optics*
Bernhard Schölkopf (Max Planck Institute Tuebingen)
[Chair: Johan Suykens]

17:30-19:00 Reception in Salons Arenberg Castle

# Program

## Tuesday July 9

09:00-09:50 *Large-scale Convex Optimization for Machine Learning*
Francis Bach (INRIA)
[Chair: Joos Vandewalle]

09:50-10:40 *Domain-Specific Languages for Large-Scale Convex Optimization*
Stephen Boyd (Stanford University)
[Chair: Joos Vandewalle]

10:40-11:00 Coffee break

11:00-12:00 Oral session 2: *Optimization algorithms*
[Chair: Joos Vandewalle]

12:00-12:20 Spotlight presentations Poster session 1 (2 min/poster)
[Chair: Vanya Van Belle]

12:20-14:30 Group picture and Lunch in De Moete
Poster session 1 in Rooms S

14:30-15:20 *Dynamic $L_1$ Reconstruction*
Justin Romberg (Georgia Tech)
[Chair: Kristiaan Pelckmans]

15:20-16:10 *Multi-task Learning*
Massimiliano Pontil (University College London)
[Chair: Andreas Argyriou]

16:10-16:30 Coffee break

16:30-18:30 Oral session 3: *Kernel methods and support vector machines*
[Chair: Marco Signoretto]

19:00 Dinner in Faculty Club

# Program

## Wednesday July 10

09:00-09:50 *Subgradient methods for huge-scale optimization problems*
Yurii Nesterov (Catholic University of Louvain)
[Chair: Andreas Argyriou]

09:50-10:40 *Living on the edge: A geometric theory of phase transitions in convex optimization*
Joel Tropp (California Institute of Technology)
[Chair: Kristiaan Pelckmans]

10:40-11:00 Coffee break

11:00-12:30 Oral session 4: *Structured low-rank approximation*
[Chair: Kristiaan Pelckmans]

12:30-12:50 Spotlight presentations Poster session 2 (2 min/poster)
[Chair: Vanya Van Belle]

12:50-14:30 Lunch in De Moete
Poster session 2 in Rooms S

14:30-15:20 *Minimum error entropy principle for learning*
Ding-Xuan Zhou (City University of Hong Kong)
[Chair: Johan Suykens]

15:20-16:10 *Learning from Weakly Labeled Data*
James Kwok (Hong Kong University of Science and Technology)
[Chair: Johan Suykens]

16:10-16:30 Coffee break

16:30-18:00 Oral session 5: *Robustness*
[Chair: Kris De Brabanter]

18:00 Closing

# Program - Oral sessions

## Oral session 1: Feature selection and sparsity
**(July 8, 15:10-16:40)**

15:10-15:40 *The graph-guided group lasso for genome-wide association studies*
Zi Wang and Giovanni Montana
Mathematics Department, Imperial College London

15:40-16:10 *Feature Selection via Detecting Ineffective Features*
Kris De Brabanter[1] and Laszlo Györfi[2]
[1] KU Leuven ESAT-SCD
[2] Dep. Comp. Sc. & Inf. Theory, Budapest Univ. of Techn. and Econ.

16:10-16:40 *Sparse network-based models for patient classification using fMRI*
Maria J. Rosa, Liana Portugal, John Shawe-Taylor and Janaina Mourao-Miranda: Computer Science Department, University College London

## Oral session 2: Optimization algorithms
**(July 9, 11:00-12:00)**

11:00-11:30 *Incremental Forward Stagewise Regression: Computational Complexity and Connections to LASSO*
Robert M. Freund[1], Paul Grigas[2] and Rahul Mazumder[2]
[1] MIT Sloan School of Management, [2] MIT Operations Research Center

11:30-12:00 *Fixed-Size Pegasos for Large Scale Pinball Loss SVM*
Vilen Jumutc, Xiaolin Huang and Johan A.K. Suykens
KU Leuven ESAT-SCD

# Program - Oral sessions

## Oral session 3: Kernel methods and support vector machines
**(July 9, 16:30-18:30)**

16:30-17:00 *Output Kernel Learning Methods*
Francesco Dinuzzo[1], Cheng Soon Ong[2] and Kenji Fukumizu[3]
[1] MPI for Intelligent Systems Tuebingen, [2] NICTA, Melbourne,
[3] Institute of Statistical Mathematics, Tokyo

17:00-17:30 *Deep Support Vector Machines for Regression Problems*
M.A. Wiering, M. Schutten, A. Millea, A. Meijster and L.R.B. Schomaker
Institute of Artif. Intell. and Cognitive Eng., Univ. of Groningen

17:30-18:00 *Subspace Learning and Empirical Operator Estimation*
Alessandro Rudi[1], Guillermo D. Canas[2] and Lorenzo Rosasco[2,3]
[1] Istituto Italiano di Tecnologia, [2] MIT-IIT,
[3] Universita di Genova

18:00-18:30 *Kernel based identification of systems with multiple outputs using nuclear norm regularization*
Tillmann Falck, Bart De Moor and Johan A.K. Suykens
KU Leuven, ESAT-SCD and iMinds Future Health

# Program - Oral sessions

## Oral session 4: Structured low-rank approximation
**(July 10, 11:00-12:30)**

11:00-11:30 *First-order methods for low-rank matrix factorization applied to informed source separation*
Augustin Lefevre[1] and Francois Glineur[1,2]
[1] ICTEAM Institute and [2] CORE Institute, Universite catholique de Louvain

11:30-12:00 *Structured low-rank approximation as optimization on a Grassmann manifold*
Konstantin Usevich and Ivan Markovsky
Dep. ELEC, Vrije Universiteit Brussel

12:00-12:30 Scalable Structured Low Rank Matrix Optimization Problems
Marco Signoretto[1], Volkan Cevher[2] and Johan A.K. Suykens[1]
[1] KU Leuven, ESAT-SCD, [2] LIONS, EPFL Lausanne

## Oral session 5: Robustness
**(July 10, 16:30-18:00)**

16:30-17:00 *Learning with Marginalized Corrupted Features*
Laurens van der Maaten[1], Minmin Chen[2], Stephen Tyree[2] and Kilian Weinberger[2]: [1] Delft University of Technology, [2] Washington Univ. St. Louis

17:00-17:30 *Robust regularized M-estimators of regression parameters and covariance matrix*
Esa Ollila, Hyon-Jung Kim and Visa Koivunen
Dep. of Signal Processing and Acoustics, Aalto University

17:30-18:00 *Robust Near-Separable Nonnegative Matrix Factorization Using Linear Optimization*, Nicolas Gillis[1] and Robert Luce[2]
[1] ICTEAM Institute, Univ. catholique de Louvain, [2] Technische Univ. Berlin

# Program - Poster sessions

## Poster session 1
**(July 9, 13:15-14:30)**

- *Data-Driven and Problem-Oriented Multiple-Kernel Learning*
  Valeriya Naumova and Sergei V. Pereverzyev
  Johann Radon Institute for Computational and Applied Mathematics (RICAM)
  Austrian Academy of Sciences, Linz

- *Support Vector Machine with spatial regularization for pixel classification*
  Remi Flamary[1] and Alain Rakotomamonjy[2]
  [1] Lagrange Lab., CNRS, Universite de Nice Sophia-Antipolis,
  [2] LITIS Lab., Universite de Rouen

- *Regularized structured low-rank approximation*
  Mariya Ishteva and Konstantin Usevich and Ivan Markovsky
  Dep. ELEC, Vrije Universiteit Brussel

- *A Heuristic Approach to Model Selection for Online Support Vector Machines*
  Davide Anguita, Alessandro Ghio, Isah A. Lawal and Luca Oneto
  DITEN, University of Genoa

- *Lasso and Adaptive Lasso with Convex Loss Functions*
  Wojciech Rejchel
  Nicolaus Copernicus University, Torun

- *Conditional Gaussian Graphical Models for Multi-output Regression of Neuroimaging Data*
  Andre F. Marquand[1], Maria Joao Rosa[2] and Orla Doyle[1]
  [1] King's College London, [2] University College London

- *High-dimensional convex optimization via optimal affine subgradient algorithms*
  Masoud Ahookhosh and Arnold Neumaier
  Faculty of Mathematics, University of Vienna

- *Joint Estimation of Modular Gaussian Graphical Models*
  Jose Sanchez and Rebecka Jörnsten
  Mathematical Sciences, Chalmers Univ, of Technology and University of Gothenburg

- *Learning Rates of $l_1$-regularized Kernel Regression*
  Lei Shi, Xiaolin Huang and Johan A.K. Suykens
  KU Leuven, ESAT-SCD

- *Reduced Fixed-Size LSSVM for Large Scale Data*
  Raghvendra Mall and Johan A.K. Suykens
  KU Leuven, ESAT-SCD

# Program - Poster sessions

## Poster session 2
**(July 10, 13:15-14:30)**

- *Pattern Recognition for Neuroimaging Toolbox*
  Jessica Schrouff[1], Maria J. Rosa[2], Jane Rondina[2], Andre F. Marquand[3], Carlton Chu[4], John Ashburner[5], Jonas Richiardi[6], Christophe Phillips[1] and Janaina Mourao-Miranda[2]
  [1] Cyclotron Research Centre, University of Liege, [2] Computer Science Dep., University College London, [3] Institute of Psychology, King's College, London, [4] NIMH, NIH, Bethesda, [5] Wellcome Trust Centre for Neuroimaging, University College London, [6] Stanford University

- *Stable LASSO for High-Dimensional Feature Selection through Proximal Optimization*
  Roman Zakharov and Pierre Dupont
  ICTEAM Institute, Universite catholique de Louvain

- *Regularization in topology optimization*
  Atsushi Kawamoto, Tadayoshi Matsumori, Daisuke Murai and Tsuguo Kondoh
  Toyota Central R&D Labs., Inc., Nagakute

- *Classification of MCI and AD patients combining PET data and psychological scores*
  Fermin Segovia, Christine Bastin, Eric Salmon and Christophe Phillips
  Cyclotron Research Centre, University of Liege

- *Kernels design for Internet traffic classification*
  Emmanuel Herbert[1], Stephane Senecal[1] and Stephane Canu[2]
  [1] Orange Labs, Issy-les-Moulineux, [2] LITIS/INSA, Rouen

- *Kernel Adaptive Filtering: Which Technique to Choose in Practice*
  Steven Van Vaerenbergh and Ignacio Santamaria
  Dep. of Communications Engineering, University of Cantabria

- *Structured Machine Learning for Mapping Natural Language to Spatial Ontologies*
  Parisa Kordjamshidi and Marie-Francine Moens
  Dep. of Computer Science, Katholieke Universiteit Leuven

- *Windowing strategies for on-line multiple kernel regression*
  Manuel Herrera and Rajan Filomeno Coelho
  BATir Dep., Universite libre de Bruxelles

- *Non-parallel semi-supervised classification*
  Siamak Mehrkanoon and Johan A.K. Suykens
  KU Leuven, ESAT-SCD

- *Visualisation of neural networks for model reduction*
  Tamas Kenesei and Janos Abonyi
  Dep. of Process Engineering, University of Pannonia

- *Convergence analysis of stochastic gradient descent on strongly convex objective functions*
  Cheng Tang and Claire Monteleoni
  Dep. of Computer Science, The George Washington University

# Invited talk abstracts

# Deep-er Kernels

**John Shawe-Taylor**

Centre for Computational Statistics and Machine Learning
University College London

**Abstract**: Kernels can be viewed as shallow in that learning is only applied in a single (output) layer. Recent successes with deeper networks highlight the need to consider richer function classes. The talk will review and discuss methods that have been developed to enable richer kernel classes to be learned. While some of these methods rely on greedy procedures many are supported by statistical learning analyses and/or convergence bounds. The talk will highlight the potential for further research on this topic.

# Connections between the Lasso and Support Vector Machines

**Martin Jaggi**

CMAP, Ecole Polytechnique, Paris

**Abstract**: We discuss the relation of two fundamental tools in machine learning and signal processing, that is the support vector machine (SVM) for classification, and the Lasso technique used in regression. By outlining a simple equivalence between the Lasso primal and the SVM dual problem, we argue that many existing optimization algorithms can also be applied to the respective other task, and that many known theoretical insights can be translated between the two settings. One such consequence is that the sparsity of a Lasso solution is equal to the number of support vectors for the corresponding SVM instance, and that one can use screening rules to prune the set of support vectors. Another example is a kernelized version of the Lasso, analogous to the kernel trick in the SVM setting. On the algorithms side, we will discuss popular greedy first-order methods used in both settings.

# Kernel Mean Embeddings applied to Fourier Optics

**Bernhard Schölkopf**

Empirical Inference Department
Max Planck Institute for Intelligent Systems

# Large-scale Convex Optimization for Machine Learning

**Francis Bach**

INRIA - SIERRA project-team
Laboratoire d'Informatique de l'Ecole Normale Superieure, Paris

**Abstract**: Many machine learning and signal processing problems are traditionally cast as convex optimization problems. A common difficulty in solving these problems is the size of the data, where there are many observations ("large n") and each of these is large ("large p"). In this setting, online algorithms which pass over the data only once, are usually preferred over batch algorithms, which require multiple passes over the data. In this talk, I will present several recent results, showing that in the ideal infinite-data setting, online learning algorithms based on stochastic approximation should be preferred, but that in the practical finite-data setting, an appropriate combination of batch and online algorithms leads to unexpected behaviors, such as a linear convergence rate with an iteration cost similar to stochastic gradient descent. (joint work with Nicolas Le Roux, Eric Moulines and Mark Schmidt)

# Domain-Specific Languages for Large-Scale Convex Optimization

**Stephen Boyd**

Information Systems Laboratory, Stanford University

**Abstract**: joint work with Eric Chu - Specialized languages for describing convex optimization problems, and associated parsers that automatically transform them to canonical form, have greatly increased the use of convex optimization in applications, especially those where the problem instances are not very large scale. CVX and YALMIP, for example, allow users to rapidly prototype applications based on solving (modest size) convex optimization problems. More recently, similar techniques were used in CVXGEN to automatically generate super-efficient small footprint code for solving families of small convex optimization optimization problems, as might be used in real-time control. In this talk I will describe the general methods used in such systems, and describe methods by which they can be adapted for large-scale problems.

# Dynamic $L_1$ Reconstruction

**Justin Romberg**

School of Electrical and Computer Engineering
Georgia Tech

**Abstract**: Sparse signal recovery often involves solving an L1-regularized optimization problem. Most of the existing algorithms focus on the static settings, where the goal is to recover a fixed signal from a fixed system of equations. This talk will have two parts. In the first, we present a collection of homotopy-based algorithms that dynamically update the solution of the underlying L1 problem as the system changes. The sparse Kalman filter solves an L1-regularized Kalman filter equation for a time-varying signal that follows a linear dynamical system. Our proposed algorithm sequentially updates the solution as the new measurements are added and the old measurements are removed from the system.
In the second part of the talk, we will discuss a continuous time "algorithm" (i.e. a set of coupled nonlinear differential equations) for solving a class of sparsity regularized least-square problems. We characterize the convergence properties of this neural-net type system, with a special emphasis on the case when the final solution is indeed sparse.
This is joint work with M. Salman Asif, Aurele Balavoine, and Chris Rozell

# Multi-task Learning

**Massimiliano Pontil**

Centre for Computational Statistics and Machine Learning
University College London

**Abstract**: A fundamental limitation of standard machine learning methods is the cost incurred by the preparation of the large training samples required for good generalization. A potential remedy is offered by multi-task learning: in many cases, while individual sample sizes are rather small, there are samples to represent a large number of learning tasks (linear regression problems), which share some constraining or generative property. If this property is sufficiently simple it should allow for better learning of the individual tasks despite their small individual sample sizes. In this talk I will review a wide class of multi-task learning methods which encourage low-dimensional representations of the regression vectors. I will describe techniques to solve the underlying optimization problems and present an analysis of the generalization performance of these learning methods which provides a proof of the superiority of multi-task learning under specific conditions.

# Subgradient methods for huge-scale optimization problems

**Yurii Nesterov**

CORE, Catholic University of Louvain, Louvain-la-Neuve

**Abstract**: We consider a new class of huge-scale problems, the problems with sparse subgradients. The most important functions of this type are piece-wise linear. For optimization problems with uniform sparsity of corresponding linear operators, we suggest a very efficient implementation of subgradient iterations, which total cost depends logarithmically on the dimension. This technique is based on a recursive update of the results of matrix/vector products and the values of symmetric functions. It works well, for example, for matrices with few nonzero diagonals and for max-type functions. We show that the updating technique can be efficiently coupled with the simplest subgradient methods, the unconstrained minimization method by Polyak, and the constrained minimization scheme by Shor. Similar results can be obtained for a new non- smooth random variant of a coordinate descent scheme. We discuss an extension of this technique onto conic optimization problems.

# Living on the edge: A geometric theory of phase transitions in convex optimization

**Joel Tropp**

Department of Computing and Mathematical Sciences
California Institute of Technology

**Abstract**: Recent empirical research indicates that many convex optimization problems with random constraints exhibit a phase transition as the number of constraints increases. For example, this phenomenon emerges in the l1 minimization method for identifying a sparse vector from random linear samples. Indeed, this approach succeeds with high probability when the number of samples exceeds a threshold that depends on the sparsity level; otherwise, it fails with high probability.

This talk summarizes a rigorous analysis that explains why phase transitions are ubiquitous in random convex optimization problems. It also describes tools for making reliable predictions about the quantitative aspects of the transition, including the location and the width of the transition region. These techniques apply to regularized linear inverse problems with random measurements, to demixing problems under a random incoherence model, and also to cone programs with random affine constraints.

Joint work with D. Amelunxen, M. Lotz, and M. B. McCoy.

# Minimum error entropy principle for learning

**Ding-Xuan Zhou**

Department of Mathematics
City University of Hong Kong

**Abstract**: Information theoretical learning is inspired by introducing information theory ideas into a machine learning paradigm. Minimum error entropy is a principle of information theoretical learning and provides a family of supervised learning algorithms. It is a substitution of the classical least squares method when the noise is non-Gaussian. Its idea is to extract from data as much information as possible about the data generating systems by minimizing error entropies. In this talk we will discuss some minimum error entropy algorithms in a regression setting by minimizing empirical Renyi's entropy of order 2. Consistency results and learning rates are presented. In particular, some error estimates dealing with heavy-tailed noise will be given.

# Learning from Weakly Labeled Data

**James Kwok**

Department of Computer Science and Engineering
The Hong Kong University of Science and Technology Hong Kong

**Abstract**: In many machine learning problems, the labels of the training examples are incomplete. These include, for example, (i) semi-supervised learning where labels are partially known; (ii) multi-instance learning where labels are implicitly known; and (iii) clustering where labels are completely unknown. In this talk, focusing on the SVM as the learner, I will describe a label generation strategy that leads to a convex relaxation of the underlying mixed integer programming problem. Computationally, it can be solved via a sequence of SVM subproblems that are much more scalable than other convex SDP relaxations. Empirical results on the three weakly labeled learning tasks above also demonstrate improved performance. (joint work with Yu-Feng Li, Ivor W. Tsang, and Zhi-Hua Zhou)

# Oral and poster session abstracts

# An Equivalence between the Lasso and Support Vector Machines

Martin Jaggi
CMAP, École Polytechnique, Palaiseau, France
jaggi@cmap.polytechnique.fr

**Abstract**: We investigate the relation of two fundamental tools in machine learning and signal processing, that is the support vector machine (SVM) for classification, and the Lasso technique used in regression. We show [7] that the resulting optimization problems are equivalent, in the following sense: Given any instance of one of the two problems, we construct an instance of the other, having the same optimal solution.
In consequence, many existing optimization algorithms for both SVMs and Lasso can also be applied to the respective other problem instances. Also, the equivalence allows for many known theoretical insights for SVM and Lasso to be translated between the two settings. One such implication gives a simple kernelized version of the Lasso, analogous to the kernels used in the SVM setting. Another consequence is that the sparsity of a Lasso solution is equal to the number of support vectors for the corresponding SVM instance, and that one can use screening rules to prune the set of support vectors. Furthermore, we can relate sublinear time algorithms for the two problems, and give a new such algorithm variant for the Lasso.

**Keywords**: Lasso, SVM, Kernel Methods, $\ell_1$-Regularized Least Squares, Screening Rules

## 1 Introduction

Large margin classification and kernel methods, and in particular the support vector machine (SVM) [3], are among the most popular standard tools for classification. On the other hand, $\ell_1$-regularized least squares regression, i.e. the Lasso estimator [8], is one of the most widely used tools for robust regression and sparse estimation. However, the two research topics developed largely independently and were not much set into context with each other.

**Support Vector Machines.** In this work, we focus on SVM large margin classifiers whose dual optimization problem is of the form

$$\min_{x \in \triangle} \|Ax\|_2^2 \ . \tag{1}$$

Here the matrix $A \in \mathbb{R}^{d \times n}$ contains all $n$ datapoints as its columns, and $\triangle$ is the unit simplex in $\mathbb{R}^n$, i.e. the set of probability vectors. The formulation (1) includes the commonly used soft-margin SVM with $\ell_2$-loss (for one or two classes, with regularized or no offset, with or without using a kernel), as given by

$$\begin{aligned} \min_{\substack{\bar{w} \in \mathbb{R}^d,\ \rho \in \mathbb{R}, \\ \xi \in \mathbb{R}^n}} \quad & \tfrac{1}{2} \|\bar{w}\|_2^2 - \rho + \tfrac{C}{2} \textstyle\sum_i \xi_i^2 \\ s.t. \quad & y_i \cdot \bar{w}^T X_i \geq \rho - \xi_i \quad \forall i \in [1..n] \ . \end{aligned} \tag{2}$$

here $C > 0$ is the regularization parameter. (Note that optionally, this also allows the use of a regularised *offset* variable $b \in \mathbb{R}$ to obtain a classifier that does not necessarily pass through the origin, by the trick of increasing the dimensionality of $X_i$ and $w$ by one.)

**Lasso.** On the other hand, the Lasso [8], is given by the quadratic program

$$\min_{x \in \blacklozenge} \|Ax - b\|_2^2 \ , \tag{3}$$

also known as the constrained variant of $\ell_1$-regularized least squares regression. Here the right hand side $b$ is a fixed vector $b \in \mathbb{R}^d$, and $\blacklozenge$ is the $\ell_1$-unit-ball in $\mathbb{R}^n$. Note that if the desired $\ell_1$-regularization constraint is not $\|x\|_1 \leq 1$, but $\|x\|_1 \leq r$ for some $r > 0$ instead, then it is enough to simply re-scale the input matrix $A$ by a factor of $\frac{1}{r}$, in order to obtain our above formulation (3) for any general Lasso problem. In applications of the Lasso, it is important to distinguish two alternative interpretations of the data matrix $A$, which defines the problem instance (3): On one hand, in the setting of *sparse regression*, the goal is to approximate the single vector $b$ by a combination of few dictionary vectors, being the columns of $A$, called the dictionary matrix. On the other hand if the Lasso problem is interpreted as *feature-selection*, then each row $A_{i:}$ of $A$ is interpreted as an input vector, and for each of those, the Lasso is approximating the response $b_i$ to input $A_{i:}$. See e.g. [1] for a recent overview of Lasso-type methods.

**Related Work.** The early work of [6] has already significantly deepened the joint understanding of kernel methods and the sparse coding setting of the Lasso. However, despite its title, [6] is not addressing SVM classifiers, but in fact the $\varepsilon$-insensitive loss variant of support vector regression (SVR). In the application paper [5], the authors already suggested to make use of the

"easier" direction of our reduction, reducing the Lasso to a very particular SVM instance, but not addressing the SVM regularization parameter.

## 2 The Equivalence

We prove that the two problems (1) and (3) are indeed equivalent, by constructing instances of the respective other problem, having the same solutions.

**2.1 (Lasso $\preceq$ SVM): Given a Lasso instance, construct an equivalent SVM instance**
In order to represent the $\ell_1$-ball $\blacklozenge$ by a simplex $\triangle$, the standard concept of *barycentric* coordinates comes to help, stating that every polytope can be represented as the convex hull of its vertices [9], which in the $\ell_1$-ball case are $\{\pm\mathbf{e}_i \mid i \in [1..n]\}$. Given a Lasso instance of the form (3), that is, $\min_{x\in\blacklozenge} \|Ax-b\|_2^2$, we can therefore directly parameterize the $\ell_1$-ball by the $2n$-dimensional simplex. By writing $(\mathbf{I}_n \,|{-}\mathbf{I}_n)x_\triangle$ for any $x \in \blacklozenge$, the objective function becomes $\|(A\,|{-}A)x_\triangle - b\|_2^2$. This means we have obtained the equivalent non-negative regression problem over the domain $x_\triangle \in \triangle \subset \mathbb{R}^{2n}$ which, by translation, is equivalent to the (hard-margin) SVM formulation (1), i.e. $\min_{x_\triangle\in\triangle} \|\tilde{A}x_\triangle\|_2^2$ , where the data matrix is given by $\tilde{A} := (A\,|{-}A) - b\mathbf{1}^T \in \mathbb{R}^{d\times 2n}$. This reduction gives us a one-to-one correspondence of all feasible solutions, preserving the objective values: For any feasible solution $x \in \mathbb{R}^n$ to the Lasso, we have a feasible SVM solution $x_\triangle \in \mathbb{R}^{2n}$ of the *same* objective value, and vice versa.

**2.2 (SVM $\preceq$ Lasso): Given an SVM instance, constructing an equivalent Lasso instance**
This reduction is harder to accomplish than the other direction we explained above. Given an instance of a (soft- or hard-margin) SVM problem (1), we suppose that we have a (possibly non-optimal) weakly-separating vector $w \in \mathbb{R}^d$ available. Given $w$, we define a Lasso instance $(\tilde{A}, \tilde{b})$ as the translated datapoints $\tilde{A} := \left\{ A_i + \tilde{b} \;\middle|\; i \in [1..n] \right\}$ together with the right hand side $\tilde{b} := -C\frac{w}{\|w\|_2}$, that is a translation in direction of $w$ for some $C > 0$. Details are given in the full paper [7], showing that a weakly-separating vector $w$ is trivial to obtain for the $\ell_2$-loss soft-margin SVM (2), even if the SVM input data is not separable.

## 3 Implications & Remarks

**3.1 Some Implications for the Lasso**
**Sublinear Time Algorithms.** Using our reduction, we observe that the recent breakthrough SVM algorithm of [2] can also be applied to the Lasso, returning an $\varepsilon$-accurate solution to (3) in time $O(\varepsilon^{-2}(n+d)\log n)$.

**A Lasso in Kernel Space.** Using our reduction, we can kernelize the Lasso fully analogously to the classical kernel trick for SVMs, resulting in the formulation $\min_{x\in\blacklozenge} \|\sum_i \Psi(A_i)x_i - \Psi(b)\|_{\mathcal{H}}^2$ . In the light of the success of the kernel idea for classification with its existing well-developed theory, it will be interesting to relate these results to the above proposed kernelized version of the Lasso, and to study how different kernels will affect the solution $x$ for applications of the Lasso.

**3.2 Some Implications for SVMs**
**Structure and Sparsity of the Support Vectors, in the View of Lasso Sparsity.** There has been a very significant boost of new literature studying the sparsity of solutions to the Lasso and related $\ell_1$-regularized methods, in particular the study of the sparsity of $x$ when $A$ and $b$ are from distributions with certain properties. Using our construction of the equivalent Lasso instance for a given SVM, such results then directly apply to the sparsity pattern of the solution to our original SVM (i.e. the pattern and the number of support vectors). More precisely, any distribution of matrices $A$ and corresponding $b$ for which the Lasso sparsity is well characterized, will also give the same patterns of support vectors for the equivalent SVM (and in particular the same number of support vectors).

**Screening Rules for Support Vector Machines.** For the Lasso, *screening rules* have been developed recently. This approach consists of a single pre-processing pass though the data $A$, in order to immediately discard those predictors $A_i$ that can be guaranteed to be inactive for the optimal solution [4]. Translated to the SVM setting by our reduction, any such Lasso screening rule can be used to permanently discard input points before the SVM optimization is started.

## References


[1] P Bühlmann and S van de Geer. *Statistics for High-Dimensional Data - Methods, Theory and Applications*. Springer, 2011.

[2] K L Clarkson, E Hazan, and D P Woodruff. Sublinear Optimization for Machine Learning. *FOCS*, 2010.

[3] C Cortes and V Vapnik. Support-Vector Networks. *Machine Learning*, 20(3):273–297, 1995.

[4] L El Ghaoui, V Viallon, and T Rabbani. Safe Feature Elimination for the LASSO and Sparse Supervised Learning Problems. *arXiv.org*, cs.LG, 2010.

[5] D Ghosh and A M Chinnaiyan. Classification and Selection of Biomarkers in Genomic Data Using LASSO. *Journal of Biomedicine and Biotechnology*, 2005(2):147–154, 2005.

[6] F Girosi. An Equivalence Between Sparse Approximation and Support Vector Machines. *Neural Computation*, 10(6):1455–1480, 1998.

[7] M Jaggi. An Equivalence between the Lasso and Support Vector Machines. arXiv, 2013.

[8] R Tibshirani. Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[9] G M Ziegler. *Lectures on Polytopes*, volume 152 of *Graduate Texts in Mathematics*. Springer, 1995.

# The graph-guided group lasso for genome-wide association studies

Zi Wang
Mathematics Department, Imperial College
180 Queen's Gate, London SW7 2AZ
zi.wang11@imperial.ac.uk

Giovanni Montana
Mathematics Department, Imperial College
180 Queen's Gate, London SW7 2AZ
g.montana@imperial.ac.uk

**Abstract**: In this work we propose a penalised regression model in which the covariates are known to be clustered into groups, and the clusters are arranged as nodes in a graph. We are motivated by an application to genome-wide association studies in which the objective is to identify important predictors, single nucleotide polymorphisms (SNPs), that account for the variability of a quantitative trait. In this applications, SNPs naturally cluster into SNP sets representing genes, and genes are treated as nodes of a biological network encoding the functional relatedness of genes. Our proposed graph-guided group lasso (GGGL) takes into account such prior knowledge available on the covariates at two different levels, and allows to select important SNPs sets while also favouring the selection of functionally related genes. We describe a computationally efficient algorithm for parameter estimation, provide experimental results and present a GWA study on lipids levels in two Asian populations.

**Keywords**: sparse group lasso, Laplacian penalty, genome-wide association studies

## 1 Introduction

Genome-wide association studies (GWAs) are concerned with the search of common genetic variants across the human genome that are associated to a disease status or quantitative trait. The genetic markers are often taken to be single-nucleotide polymorphisms (SNPs), and are treated as covariates in a linear regression model in which the response is a continuous measurement. We let $X$ be the $n \times p$ design matrix containing $n$ independent samples for which $p$ SNPs have been observed, and $y$ be the $n$-dimensional vector containing the univariate quantitative traits. We further assume that $X$ and $y$ are column-wise normalized to have sum zero and unit length.

Since the objective is to carry our variable selection, the empirical loss is minimised subject to some constraint conditions placed on the coefficients, in order to regularised the solution and carry out variable selection [1, 2].

One way of improving variable selection accuracy in GWAs is to make use of available prior knowledge about the genetic markers and the functional relationships between genes. Such knowledge typically includes the grouping of SNPs into genes, and it has been observed that selecting groups of SNPs in a single block, rather than individual SNPs in isolation, may increase the power to detect true causative and rare variants (e.g. [3]). However, additional information can also be obtained from publicly available data bases in the form of biological networks encoding pairwise interactions between genes or proteins associated to those genes. Under the assumption that such networks describe true biological processes, there are reasons to be believe that using this additional information to guide the SNP selection process may produce results that are biologically more plausible and easy to interpret as well as increase. Regularised regression models that take into consideration the integrated effects of all SNPs that belong to functionally related genes are also believed to achieve superior performance in terms of detecting the true causative markets [4, 5]. In previous GWA studies, this has been accomplished by using variations of the group lasso [6] and the sparse group lasso [7]. When groups overlap, for instance when a SNP is mapped to more than one single gene, variables selected by the overlapping group lasso [8] are the union of some groups.

To the best of our knowledge, graph structures places on genetic markers or genes are not yet used to drive the variable selection process in GWA studies with quantitative traits. In a typical gene network, two nodes are connected by an edge if the associated genes belong to the same genetic pathway or are deemed to share related functions. In the case of a weighted graph, the weights may be a probability measure of the uncertainty of the link between the genes. In this work we consider the case where prior knowledge is available at two different levels: SNPs are grouped into genes, and a weighted gene network encodes the functional relatedness of the all pairs of genes. We propose a penalised regression model, the graph-guided group lasso,

which selects important SNP groups, while also fusing information between adjacent SNPs groups in the given biological network.

## 2 Graph-guided group lasso

Suppose that the $p$ available SNPs are grouped into mutually exclusive genes $\{R_1, R_2, ..., R_r\}$. The size of a group $R_l$ is denoted by $|R_l|$. We let $X_{R_l}$ denote the $n \times |R_l|$ matrix where the columns correspond to SNPs in $R_l$, and $\mathcal{G} = \mathcal{G}(V, E)$ the gene network with vertex set $V$ corresponding to the $r$ genes in $\mathcal{R}$. The weight of the edge $k - l$ is denoted by $w_{kl}$. For simplicity, we assume that all the weights are non-negative.

The regression coefficients are obtaining by minimising $\|y - X\beta\|_2^2$ plus a penalty term given by

$$2\lambda_1 \sum_{g=1}^{r} \sqrt{|R_g|}\|\beta_{R_g}\|_2 + 2\lambda_2\|\beta\|_1 + \mu \sum_{i \in R_k, j \in R_l, R_k \sim R_l} w_{kl}(\beta_i - \beta_j)^2 \quad (1)$$

where $\lambda_1$, $\lambda_2$, and $\mu$ are non-negative regularization parameters, and $R_k \sim R_l$ if and only if they are connected in the network $\mathcal{G}$. This model has two main features. Firstly, by making use of the Laplacian penalty on the complete bipartite graph $(R_i, R_j)$ for all $R_j \sim R_i$, information is fused from all other genes interacting with $R_i$ in $\mathcal{G}$ so that these functionally related genes are encouraged to be selected in and out of the model altogether. ([4]) Secondly, there is a grouping effect, in the sense that all SNPs within a gene $R_i$ are either selected together or not selected. This feature follows from the properties of the sparse group lasso penalty [7], in which sparsity of genes and SNPs are regularized by $\lambda_1$ and $\lambda_2$ respectively.

Note the prior knowledge represented by grouping and the graph are at heterogeneous levels, hence how the pairwise relations at genes' level influence variable selection for individual SNPs may have different answers. The proposed penalty has also the effect of smoothing the regression coefficients corresponding to all SNPs that belong to interacting genes. When $\mu \to \infty$, all these coefficients are expected to be equal.

In some cases, a modification of the model above may be preferred. If two genes are directly connected in $\mathcal{G}$, it may be preferred to encourage them to be selected or discarded altogether without smoothing the individual SNP coefficients within a gene. For this reason, we also propose a second version of the GGGL model by replacing the last term in (1) by:

$$\mu \sum_{R_k \sim R_l} w_{kl}(\bar{\beta}_{R_k} - \bar{\beta}_{R_l})^2 \quad (2)$$

where $\bar{\beta}_{R_k}$ denotes the average coefficient for predictors in $R_k$. We show that, using (2), the interacting genes are indeed encouraged to be selected in or out of the model altogether, nonetheless no smoothing effect is imposed on the coefficients corresponding to the SNPs within a gene. In summary, the penalty (1) is more desirable when the interest is only in selecting genes, regardless of the specific SNP effects that drive the gene selection process, whereas (2) is more appropriate for the detection of localised SNP effects.

In summary, we propose a sparse regression model, graph-driven group lasso, for GWA studies that allows to incorporate prior knowledge at two different levels. We describe a computationally efficient estimation algorithm for both version of the model, which is based on coordinate descent methods. We also carry out extensive power studies using realistically simulated data, and compare the proposed model to the original group lasso [6] and a regression model with a network constrained penalty [4]. Finally, we present a real application to detect genetic effects associated to lipids levels in two Asian cohorts.

## References


[1] Tibshirani. Regression shrinkage and selection via the lasso. *J.R.Statist.* Soc.B, 58:267-288. 1996.

[2] Wu *et al.* Genome-wide association analysis by lasso penalized logistic regression. *Bioinformatics* 25(6):714-721. 2009.

[3] Zhou *et al.* Association screening of common and rare genetic variants by penalized regression. *Bioinformatics* 26(19): 2375-2382. 2010.

[4] Li and Li. Network-constrained regularization and variable selection for analysis of genomic data. *Bioinformatics.* Vol. 24 no. 9, pages 1175-1182 2008.

[5] Silver and Montana. Fast identification of biological pathways associated with a quantitative trait using group lasso with overlaps. vol. 11, issue 1, article *Statistical Applications in Genetics and Molecular Biology.* vol. 11, issue 1, article 7. 2012.

[6] Yuan and Lin. Model selection and estimation in regression with grouped variables. *J.R.Statist.* Soc.B, 68(1):49-67, 2006.

[7] Friedman *et al.* A note on the group lasso and a sparse group lasso. *arXiv:*1001.0736. 2010.

[8] Jacob *et al.* Group Lasso with overlap and graph Lasso. *International Conference on Machine Learning (ICML 26)* 2009.

# Feature Selection via Detecting Ineffective Features

Kris De Brabanter
Dep. Electrical Engineering
Kasteelpark Arenberg 10, 3001 Leuven
Katholieke Universiteit Leuven
kris.debrabanter@esat.kuleuven.be

László Györfi
Dep. Computer Science & Information Theory
Magyar Tudósok körútja 2., Budapest
Budapest University of Technology and Economics
gyorfi@szit.bme.hu

**Abstract**: Consider the regression problem with a response variable $Y$ and with a feature vector $\mathbf{X}$. For the regression function $m(\mathbf{x}) = \mathbf{E}\{Y \mid \mathbf{X} = \mathbf{x}\}$, we introduce a new and simple estimator of the minimum mean squared error $L^* = \mathbf{E}\{(Y - m(\mathbf{X}))^2\}$. Let $\mathbf{X}^{(-k)}$ be the feature vector, in which the $k$-th component of $\mathbf{X}$ is missing. In this paper we analyze a nonparametric test for the hypothesis that the $k$-th component is ineffective, i.e., $\mathbf{E}\{Y \mid \mathbf{X}\} = \mathbf{E}\{Y \mid \mathbf{X}^{(-k)}\}$ a.s.

**Keywords**: feature selection, minimum mean squared error, hypothesis test

## 1 Introduction

Let the label $Y$ be a real valued random variable and let the feature vector $\mathbf{X} = (X_1, \ldots, X_d)$ be a $d$-dimensional random vector. The regression function $m$ is defined by

$$m(\mathbf{x}) = \mathbf{E}\{Y \mid \mathbf{X} = \mathbf{x}\}.$$

The minimum mean squared error, called also variance of the residual $Y - m(\mathbf{X})$, is denoted by

$$L^* := \mathbf{E}\{(Y - m(\mathbf{X}))^2\} = \min_f \mathbf{E}\{(Y - f(\mathbf{X}))^2\}.$$

The regression function $m$ and the minimum mean squared error $L^*$ cannot be calculated when the distribution of $(\mathbf{X}, Y)$ is unknown. Assume, however, that we observe data

$$D_n = \{(\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_n, Y_n)\}$$

consisting of independent and identically distributed copies of $(\mathbf{X}, Y)$. $D_n$ can be used to produce an estimate of $L^*$. Nonparametric estimates of the minimum mean squared error are given in [2, 4].

## 2 New estimate of the minimum mean squared error

One can derive a new and simple estimator of $L^*$ by considering the definition

$$L^* = \mathbf{E}\{(Y - m(\mathbf{X}))^2\} = \mathbf{E}\{Y^2\} - \mathbf{E}\{m(\mathbf{X})^2\}. \quad (1)$$

The first and second term on the right-hand-side of (1) can be estimated by $\frac{1}{n}\sum_{i=1}^n Y_i^2$ and $\frac{1}{n}\sum_{i=1}^n Y_i Y_{n,i,1}$ respectively where $Y_{n,i,1}$ denotes the labels of the first nearest neighbors of $\mathbf{X}_i$ among $\mathbf{X}_1, \ldots, \mathbf{X}_{i-1}, \mathbf{X}_{i+1}, \ldots, \mathbf{X}_n$. Therefore, the minimum mean squared error $L^*$ can be estimated by

$$\tilde{L}_n := \frac{1}{n}\sum_{i=1}^n Y_i^2 - \frac{1}{n}\sum_{i=1}^n Y_i Y_{n,i,1}. \quad (2)$$

One can show without any conditions that

$$\tilde{L}_n \to L^*$$

a.s. Moreover, for bounded $|Y|$ and $\|X\|$, and for Lipschitz continuous $m$, and for $d \geq 2$, we have (cf. [3])

$$\mathbf{E}\{|\tilde{L}_n - L^*|\} \leq c_1 n^{-1/2} + c_2 n^{-2/d}.$$

## 3 Feature Selection and Hypothesis Test

One way of feature selection would be to detect ineffective components of the feature vector. Let $\mathbf{X}^{(-k)} = (X_1, \ldots, X_{k-1}, X_{k+1}, \ldots, X_d)$ be the $d-1$ dimensional feature vector such that we leave out the $k$-th component from $\mathbf{X}$. Then the corresponding minimum error is

$$L^{*(-k)} := \mathbf{E}\big\{Y - \mathbf{E}\{Y|\mathbf{X}^{(-k)}\}\big\}^2.$$

We want to test the following (null) hypothesis:

$$\mathcal{H}_k : L^{*(-k)} = L^*,$$

which means that leaving out the $k$-th component the minimum mean squared error does not increase. The hypothesis $\mathcal{H}_k$ means that

$$m(\mathbf{X}) = \mathbf{E}\{Y \mid \mathbf{X}\} = \mathbf{E}\{Y \mid \mathbf{X}^{(-k)}\} =: m^{(-k)}(\mathbf{X}^{(-k)})$$

a.s.

By using the data

$$D_n^{(-k)} = \{(\mathbf{X}_1^{(-k)}, Y_1), \ldots, (\mathbf{X}_n^{(-k)}, Y_n)\},$$

$L^{*(-k)}$ can be estimated by

$$\tilde{L}_n^{(-k)} := \frac{1}{n}\sum_{i=1}^{n} Y_i^2 - \frac{1}{n}\sum_{i=1}^{n} Y_i Y_{n,i,1}^{(-k)},$$

so the corresponding test statistic is

$$\tilde{L}_n^{(-k)} - \tilde{L}_n = \frac{1}{n}\sum_{i=1}^{n} Y_i (Y_{n,i,1} - Y_{n,i,1}^{(-k)}).$$

We can accept the hypothesis $\mathcal{H}_k$ if

$$\tilde{L}_n^{(-k)} - \tilde{L}_n$$

is "close" to zero. Since with large probability the first nearest neighbors of $\mathbf{X}_i$ and of $\mathbf{X}_i^{(-k)}$ are the same, $Y_{n,i,1} - Y_{n,i,1}^{(-k)} = 0$ in the test statistic. We know that $\mathbf{P}(Y_{n,i,1} = Y_{n,i,1}^{(-k)})$ is decreasing as $n$ increases (and $d$ remains fixed) and vice versa, this probability is increasing as $d$ increases (while $n$ remains fixed). Hence, this test statistic is small even when the hypothesis $\mathcal{H}_k$ is not true.

To correct for this problem we modify the test statistic such that

$$(\hat{Y}_{n,i,1}, \hat{Y}_{n,i,1}^{(-k)}) = (Y_{n,i,1}, Y_{n,i,1}^{(-k)}) \quad \text{if} \quad Y_{n,i,1} \neq Y_{n,i,1}^{(-k)}$$

and

$$(\hat{Y}_{n,i,1}, \hat{Y}_{n,i,1}^{(-k)}) = I_i(Y_{n,i,2}, Y_{n,i,1}^{(-k)}) + (1-I_i)(Y_{n,i,1}, Y_{n,i,2}^{(-k)}),$$

otherwise (where $Y_{n,i,2}$ denotes the labels of the second nearest neighbors of $\mathbf{X}_i$ among $\mathbf{X}_1, \ldots, \mathbf{X}_{i-1}, \mathbf{X}_{i+1}, \ldots, \mathbf{X}_n$), with

$$I_i = \begin{cases} 0 & \text{with probability } 1/2, \\ 1 & \text{with probability } 1/2, \end{cases}$$

yielding

$$\hat{L}_n^{(-k)} - \hat{L}_n = \frac{1}{n}\sum_{i=1}^{n} Y_i (\hat{Y}_{n,i,1} - \hat{Y}_{n,i,1}^{(-k)}).$$

As in classical hypothesis testing, we need to find the limit distribution of the test statistic. The main difficulty here is that $\hat{L}_n^{(-k)} - \hat{L}_n$ is an average of dependent random variables. However, this dependence has a special property, called exchangeable. Based on a central limit theorem for exchangeable arrays [1], we can show the following result.

**Theorem 1** *Under the conditions of [1, Theorem 2], we have that*

$$\sqrt{n}(\hat{L}_n^{(-k)} - \hat{L}_n) \xrightarrow{d} N(0, 2L^* \mathbf{E}\{Y^2\})$$

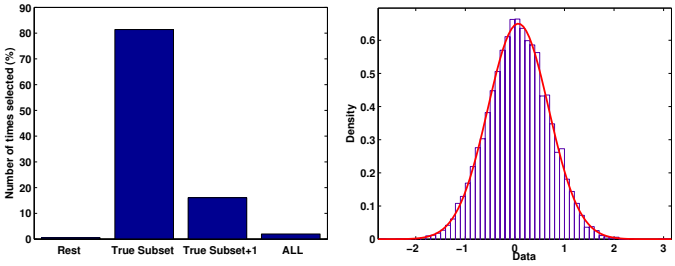
*under the null hypothesis* $\mathcal{H}_k$.

In the above theorem, $L^*$ and $\mathbf{E}\{Y^2\}$ can be estimated by (2) and $\frac{1}{n}\sum_{i=1}^{n} Y_i^2$ respectively. Note that such a results is quite surprising, since under $\mathcal{H}_k$ the smoothness of the regression function $m$ and the dimension $d$ do not count.

## 4 Simulations

First, consider the following nonlinear function with 4 uniformly distributed inputs on $[0,1]^4$ with $n = 1,000$: $Y = \sin(\pi X^{(1)})\cos(\pi X^{(4)}) + \varepsilon$, with $\varepsilon \sim N(0, 0.1^2)$. Figure 1(a) illustrates the frequency of the true selected subset, true subset with additional component and full subset selected by the proposed test procedure during 1,000 runs. The significance level is set to 0.05.

Second, we experimentally verify Theorem 1 by means of bootstrap (10,000 replications). Consider the following five dimensional function with additive noise: $Y = \sum_{i=1}^{5} c_i X^{(i)} + \varepsilon$, where $c_1 = 0$ and $c_i = 1$ for $i = 2, \ldots, 5$. Let $\mathbf{X}$ be uniform on $[0,1]^5$ and $\varepsilon \sim N(0, 0.05^2)$. Figure 1(b) shows the histogram of $\sqrt{n}(\hat{L}_n^{(-k)} - \hat{L}_n)$ under the null hypothesis i.e., $\mathcal{H}_k$ for $k = 1$. A Kolmogorov-Smirnov test confirms this result.




(a) (b)

**Fig. 1:** (a) Illustration of the frequency of the true selected subset, true subset with additional component and full subset. Rest denotes at least one component of the true subset is selected; (b) Density histogram of $\sqrt{n}(\hat{L}_n^{(-k)} - \hat{L}_n)$ under the null hypothesis with corresponding Normal fit.

## 5 Conclusion

We have presented a simple nonparametric hypothesis test for detecting ineffective features. The simulation shows the capability of the proposed methodology.

**Acknowledgments**
Kris De Brabanter is supported by an FWO fellowship grant. László Györfi was partially supported by the European Union and the European Social Fund through project FuturICT.hu (grant no.: TAMOP-4.2.2.C-11/1/KONV-2012-0013).

## References

[1] J.R. Blum, H. Chernoff, M. Rosenblatt & H. Teicher. Central limit theorems for interexchangeable processes. *Canad. J. Math.*, 10:222–2229, 1958.

[2] L. Devroye, D. Schäfer, L. Györfi & H. Walk. The estimation problem of minimum mean squared error. *Statistics and Decisions*, 21(1): 15-28, 2003.

[3] L. Devroye, P. Ferrario, L. Györfi & H. Walk. Strong universal consistent estimate of the minimum mean squared error. *Submitted*, 2013.

[4] E. Liitiäinen, F. Corona & A. Lendasse. Residual variance estimation using a nearest neighbor statistic. *J. Multivariate Anal.*, 101(4): 811-823, 2010.

# Sparse network-based models for patient classification using fMRI

Maria J. Rosa
Computer Science Department
University College London, UK
m.rosa@ucl.ac.uk

Liana Portugal
Computer Science Department
University College London, UK

John Shawe-Taylor
Computer Science Department
University College London, UK

Janaina Mourao-Miranda
Computer Science Department
University College London, UK

**Abstract**: Pattern recognition applied to whole-brain neuroimaging data, such as functional Magnetic Resonance Imaging (fMRI), has been successful at discriminating psychiatric patients from healthy subjects. However, predictive patterns obtained from whole-brain data are difficult to interpret in terms of the underlying neurobiology. As is generally accepted, most psychiatric disorders are brain connectivity disorders. Therefore, pattern recognition based on network models, in particular sparse models, should provide more scientific insight and potentially more powerful predictions than whole-brain approaches. Here, we build a novel sparse network-based discriminative modelling framework, based on Gaussian graphical models and L1-norm linear Support Vector Machines (SVM). This framework provides easier pattern interpretation, in terms of network changes, and we illustrate it by classifying patients with depression and controls, using fMRI data from a sad facial processing task.

**Keywords**: sparse models, graphical LASSO, L1-norm SVM, brain connectivity, fMRI

## 1 Introduction

Brain connectivity measures provide ways of assessing statistical relationships between signals from different brain regions [5]. These methods have revealed new insights into brain network function in general, and of network disfunction in psychiatric disorders [1]. A way of measuring connectivity is by estimating the inverse covariance (iCOV) matrix between brain regions under sparsity constraints. The zero entries in this matrix correspond to conditional independence between regions (and missing links in a Gaussian graphical model). This approach has been used for classification tasks [3] but, to our knowledge, it has not yet been combined with sparse discriminative classifiers to provide a fully sparse predictive modelling framework.

Here, we build a novel connectivity-based discriminative framework combining graphical Least Absolute Shrinkage and Selection Operator (gLASSO, [4]) and L1-norm linear SVM. We illustrate our technique by classifying patients with depression and controls, using fMRI data from a sad faces task. The resulting patterns are easier to interpret than whole-brain and non-sparse ones, by revealing a small set of connections that best discriminate between the groups.

## 2 Sparse network-based predictive modelling framework

### 2.1 Data preprocessing

For each subject, fMRI images are motion corrected and coregistered to an MNI template[1]. The images then undergo parcellation into $p$ regions from a brain atlas. Regional mean time-series are estimated by averaging the fMRI signals over all spatial elements within each region. The pairwise inter-regional covariance matrix, $\Sigma$=$[p \times p]$, is then computed from the averaged time-series.

### 2.2 Sparse graphs and predictive model

We then use gLASSO [4] to estimate a sparse (via regularisation, not thresholding) iCOV matrix for each subject, $\Omega = \Sigma^{-1}$. gLASSO tries to find $\Omega$ that maximises the penalised Gaussian log-likelihood: $\log\det\Omega - \mathrm{tr}(\Sigma\Omega) - \lambda||\Omega||_1$ using a coordinate descent optimisation procedure [4], where $\lambda$ is the regularisation parameter. The lower triangular entries of the iCOV matrices are then vectorised, divided by their norm, and used as features in a linear SVM [6] for classification. Linear L1-norm SVM solves the following optimisation problem: $\min_{\mathbf{w}} f(\mathbf{w}) \equiv ||\mathbf{w}||_1 + C\sum_{i=1}^{k} \max(1 - y_i \mathbf{w}^T \mathbf{x}_i, 0)^2$, where $C > 0$, $k$ is the number of examples, $\mathbf{x}_i \in \Re^q$

[1] Montreal Neuroimaging Institute template.

are the feature vectors and $y_i = \{-1, +1\}$ the labels (e.g. control and patient). We use a leave-one-subject-per-group-out (LOSGO) cross-validation (CV) scheme (number of folds, $n_f$, = number of subjects in each group, $n_s$), with a nested LOSGO-CV ($n_f = n_s - 1$) to independently optimise the gLASSO regularisation parameter, $\lambda$, via maximum likelihood, and the C-parameter from SVM.

### 2.3 Pattern interpretation

The weight vector, $\mathbf{w}$, is sparse, and its elements correspond to brain connections. Because each CV fold yields a different weight vector, we calculate how often each feature was selected across the repetitions. The connections that were selected at least half of the folds are referred to as the most discriminative set.

## 3 Experiments

We use the fMRI data of [2], from 19 medication-free patients with depression and 19 healthy volunteers. The paradigm involved implicit processing of sad faces of different emotional intensity (low, medium, and high). In addition to the above preprocessing, the data were smoothed in space using an 8 mm Gaussian filter. The images were parcellated into 137 regions: 122 from the BSA atlas[2] and 15 from the Harvard-Oxford atlas[3]. Our sparse network model correctly classified 74% of patients and 89% of controls, corresponding to an accuracy of 82% (p-value $< 0.05$, permutation test with 1000 samples, Table 1). The inverse covariances were on average 83% sparse (total number of connections = 9316), while the most discriminative set of connections was 99% sparse and included: amygdala and hippocampus with temporal cortex; cingulate cortex with frontal cortex and thalamus; amygdala with precentral cortex; insula with frontal cortex (Figure 1).

**Tab. 1:** Accuracies of sparse network-based framework and L2-norm SVM based whole-brain (WB) approach on the same data [2]. WB (betas) use the WB coefficients of a general linear model from a univariate analysis for classification. The * denotes a p-value $< 0.05$.

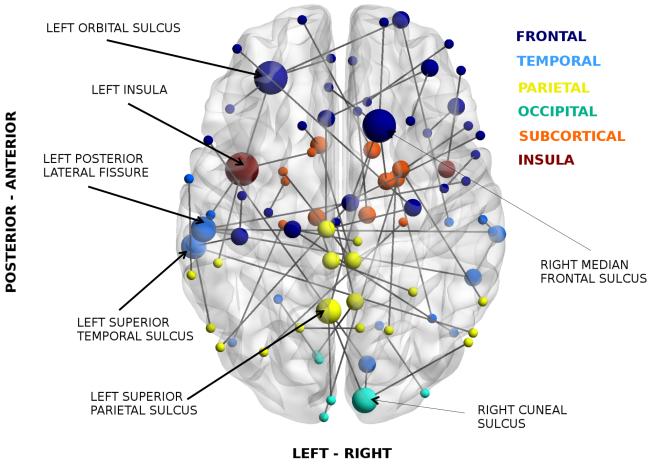
| Model | Sens. (%) | Spec. (%) | Acc. (%). |
|---|---|---|---|
| Network-based | 74 | 89 | 82* |
| WB [2] | 53 | 58 | 56 |
| WB (betas) [2] | 72 | 82 | 77* |

## 4 Discussion

Our results show that it is possible to discriminate patients with depression from controls based on sparse network-based predictive models. Compared to whole-brain analyses on the same data (using all emotional stimuli) [2], our approach provided higher accuracy, with the advantage of easier pattern interpretation, which aids the development of interpretable diagnostic tools for psychiatry. The most discriminative features are consistent with the literature [1], and highlight differences between the groups in circuitry associated with emotional regulation. Future work includes better pattern interpretation and validation with other datasets.

[2] http://lnao.lixium.fr/spip.php?article=229
[3] http://fsl.fmrib.ox.ac.uk/fsl/fslwiki/Atlases

**Fig. 1:** Sparse-networks that best discriminate patients with depression from controls. The connections shown were selected at least half of the cross-validation folds. The size of each region corresponds to the number of connected nodes.




**Acknowledgments**

This work was funded by the Wellcome Trust (UK) [WT086565/Z/08/Z].

## References

[1] A. Stuhrmann et al. Facial emotion processing in major depression: a systematic review of neuroimaging findings. *Biol Mood Anxiety Disord*, 1(1):10, 2011.

[2] C. H. Fu et al. Pattern classification of sad facial processing: toward the development of neurobiological markers in depression. *Biol. Psychiatry*, 63(7):656–662, 2008.

[3] G. Cecchi et al. Discriminative network models of schizophrenia. In *Adv. in Neural Inform. Proc. Syst. 22*, pages 252–260. 2009.

[4] J. Friedman et al. Sparse inverse covariance estimation with the graphical LASSO. *Biostatistics*, 9(3):432–441, 2008.

[5] K. Li et al. Review of methods for functional brain connectivity detection using fMRI. *Comp. Med. Im. and Graph.*, 33(2):131, 2009.

[6] R. Fan et al. LIBLINEAR: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874, 2008.

# Incremental Forward Stagewise Regression: Computational Complexity and Connections to LASSO

Robert M. Freund
MIT Sloan School of Management
rfreund@mit.edu

Paul Grigas Rahul Mazumder
MIT Operations Research Center
{pgrigas, rahulmaz}@mit.edu

**Abstract**: Incremental Forward Stagewise Regression ($\text{FS}_\varepsilon$) is a statistical algorithm that produces sparse coefficient profiles for linear regression. Using the tools of first-order methods in convex optimization, we analyze the computational complexity of $\text{FS}_\varepsilon$ and its flexible variants with adaptive shrinkage parameters. We also show that a simple modification to $\text{FS}_\varepsilon$ yields an $O(1/k)$ convergent algorithm for the least squares LASSO fit for *any* regularization parameter and *any* data-set — thereby quantitatively characterizing the nature of regularization implicitly induced by $\text{FS}_\varepsilon$.

**Keywords**: Incremental Forward Stagewise Regression, first-order methods, sparsity, $\ell_1$ regularization

## 1 Introduction

Using tools from first-order methods in convex optimization, this paper establishes connections between sparse $\ell_1$ regression [4], the Incremental Forward Stagewise (Boosting) Algorithm [10] and the notion of regularization in boosting [1], and presents new computational guarantees for these methods. We consider the linear regression model $\mathbf{y} = \mathbf{X}\beta + \mathbf{e}$, with response $\mathbf{y} \in \mathbb{R}^n$, model matrix $\mathbf{X} \in \mathbb{R}^{n\times p}$, regression coefficients $\beta \in \mathbb{R}^p$ and errors $\mathbf{e} \in \mathbb{R}^n$. In the high-dimensional statistical regime, especially with $p \gg n$, a sparse linear model with few non-zero coefficients is often desirable. In this context $\ell_1$ penalized regression, i.e., LASSO [4]:

$$\underset{\beta}{\text{minimize}} \;\; \frac{1}{2}\|\mathbf{y} - \mathbf{X}\beta\|_2^2 \;\text{ subject to }\; \|\beta\|_1 \le \delta \qquad (1)$$

is often used to perform variable selection and shrinkage in the coefficients — and is known to yield models with good predictive performance.

**Incremental Forward Stagewise:** The Incremental Forward Stagewise algorithm [1, 10] ($\text{FS}_\varepsilon$ for short) with shrinkage factor $\varepsilon$ is a type of boosting algorithm for the linear regression problem. This generates a coefficient profile[1] by repeatedly updating (by a small amount $\varepsilon$) the coefficient of the variable most correlated with the current residuals. The algorithm is initialized with residual $r^0 = \mathbf{y}$ and $\beta^0 = \mathbf{0}$, and updates at iteration $k$ as follows:

Compute $j_k \in \underset{j\in\{1,\ldots,p\}}{\arg\max} |(r^k)^T\mathbf{X}_j|$ and update:

$$\beta^{k+1}_{j_k} \leftarrow \beta^k_{j_k} + \varepsilon \text{sgn}[(r^k)^T\mathbf{X}_{j_k}] \qquad (2)$$

$$r^{k+1} \leftarrow r^k - \varepsilon \text{sgn}[(r^k)^T\mathbf{X}_{j_k}]\mathbf{X}_{j_k} \qquad (3)$$

where $\beta^k_{j_k}$ is the $j_k^{\text{th}}$ coordinate of $\beta^k$. Due to the update scheme (2), $\text{FS}_\varepsilon$ has the following desirable sparsity properties[2]:

$$\|\beta^k\|_1 \le k\varepsilon \quad \text{and} \quad \|\beta^k\|_0 \le k \;. \qquad (4)$$

In the presence of noise and specifically in the high dimensional regime, it is desirable to have a regularized solution to obtain a proper bias-variance tradeoff. A principal reason behind why $\text{FS}_\varepsilon$ (and other boosting algorithms in general) is attractive from a statistical viewpoint is because of its ability to deliver regularized solutions (4) by controlling the number of iterations along with the shrinkage parameter. Different choices of $\varepsilon$ lead to different algorithms: a choice of $\varepsilon = |(r^k)^T\mathbf{X}_{j_k}|$ in (2)-(3) yields the Forward Stagewise algorithm (FS) [1] — which is a greedy version of best-subset selection. It is therefore natural to ask what criterion does the $\text{FS}_\varepsilon$ algorithm optimize?, and what are its computational guarantees? Furthermore, since $\text{FS}_\varepsilon$ produces a coefficient profile with implicit regularization, is it possible to characterize it via a constrained least squares fit (e.g. LASSO)? To the best of our knowledge, a simple and complete answer to the above questions are heretofore unknown, apart from some special cases. It is known that *Infinitesimal* Incremental Forward Stagewise Regression ($\text{FS}_0$), the limit of $\text{FS}_\varepsilon$ as $\varepsilon \to 0+$ is the solution to a complicated differential equation [1, 10] and is in general different from the LASSO coefficient profile.

[1] A coefficient profile is a path of coefficients $\{\beta(\alpha)\}_{\alpha\in\boldsymbol{\alpha}}$ where $\alpha$ parametrizes the path. In the context of $\text{FS}_\varepsilon$, $\alpha$ indexes the $\ell_1$ arc-length of the coefficients.

[2] For a vector $x$, $\|x\|_0$ counts the number of non-zero entries.

**Our Contributions:** We derive novel complexity bounds for $\text{FS}_\varepsilon$ and its flexible variants. We also show that a simple modification to $\text{FS}_\varepsilon$ yields an $O(1/k)$ convergent algorithm for LASSO for *any* $(\mathbf{y}, \mathbf{X}, \delta)$.

## 2 $\text{FS}_\varepsilon$ as Subgradient Descent

Consider the non-smooth convex optimization problem:

$$\underset{r \in P_{\text{res}}}{\text{minimize}} \quad f(r) := \|\mathbf{X}^T r\|_\infty \tag{5}$$

where $P_{\text{res}} := \{r \in \mathbb{R}^n : r = \mathbf{y} - \mathbf{X}\beta \text{ for some } \beta \in \mathbb{R}^p\}$ is the the set of residuals. Note that (5) has optimal objective value of $f^* = 0$. We establish the following connection.

**Theorem 2.1.** *The $FS_\varepsilon$ algorithm is an instance of the Subgradient Descent Method to solve problem (5), initialized at $r^0 = \mathbf{y}$ and with a constant step-size of $\varepsilon$ at each iteration.*

Applying the computational complexity tools for the Subgradient Descent Method [3, 6], it is straightforward to establish the following computational guarantees.

**Corollary 2.1. (Complexity of $\text{FS}_\varepsilon$)** *For the constant shrinkage factor $\varepsilon$ it holds that:*

$$\min_{0 \le i \le k} \|\mathbf{X}^T r^i\|_\infty \le \frac{\|\mathbf{y}\|_2^2}{2\varepsilon(k+1)} + \frac{\varepsilon \|\mathbf{X}\|_{1,2}^2}{2} \ . \tag{6}$$

*If we a priori decide to run $FS_\varepsilon$ for $k$ iterations and set $\varepsilon := \frac{\|\mathbf{y}\|_2}{\|\mathbf{X}\|_{1,2}\sqrt{k+1}}$ then* [3]

$$\min_{0 \le i \le k} \|\mathbf{X}^T r^i\|_\infty \le \frac{\|\mathbf{X}\|_{1,2}\|\mathbf{y}\|_2}{\sqrt{k+1}} \ . \tag{7}$$

*If instead, the shrinkage factor is dynamically chosen as $\varepsilon = \varepsilon^k := \frac{|(r^k)^T \mathbf{X}_{j_k}|}{\|\mathbf{X}_{j_k}\|_2^2}$, then the bound (7) holds for all values of $k$ without having to set $k$ a priori.*

## 3 A modified $\text{FS}_\varepsilon$ for the LASSO

Suppose we modify (5) by adding a regularizing term to the objective function as follows:

$$f_\delta^* = \min_{r \in P_{\text{res}}} f_\delta(r) := \|\mathbf{X}^T r\|_\infty + \frac{1}{2\delta}\|r - \mathbf{y}\|_2^2 \ , \tag{8}$$

for some parameter $\delta > 0$. We show that a rescaled version of the above problem is a dual of the LASSO (1), thereby demonstrating one way that LASSO arises is as a quadratic regularization of (5). One method to solve (1) that has desirable sparsity properties similar to (4) is the Frank-Wolfe Method (also known as the conditional gradient method [3]). The Frank-Wolfe method with step-size sequence $\{\bar{\alpha}_k\}$, $\bar{\alpha}_k \in (0, 1)$, applied to problem (1) is initialized with residual $r^0 = \mathbf{y}$ and $\beta^0 = \mathbf{0}$, and updates at iteration $k$ as follows:

[3] Where, $\|\mathbf{X}\|_{1,2}$ is the largest $\ell_2$ norm of the columns of $\mathbf{X}$.

Compute $j_k \in \arg\max_{j \in \{1,\dots,p\}} |(r^k)^T \mathbf{X}_j|$ and update:

$$\beta^{k+1} \leftarrow (1 - \bar{\alpha}_k)\beta^k + \bar{\alpha}_k \delta \ \text{sgn}((r^k)^T \mathbf{X}_{j_k}) e_{j_k} \tag{9}$$

$$r^{k+1} \leftarrow (1 - \bar{\alpha}_k) r^k - \bar{\alpha}_k \delta \text{sgn}[(r^k)^T \mathbf{X}_{j_k}] \mathbf{X}_{j_k} \tag{10}$$

For a fixed step-size $\bar{\alpha}_k := \frac{\varepsilon}{\delta + \varepsilon}$, observe that (9) can be rearranged to:

$$\beta^{k+1} \leftarrow \frac{\delta}{\varepsilon + \delta}\left[\beta^k + \varepsilon \text{sgn}((r^k)^T \mathbf{X}_{j_k}) e_{j_k}\right] \tag{11}$$

which is equivalent to the $\text{FS}_\varepsilon$ update (2) modulo a multiplicative factor which keeps the coefficient profile within $\{\beta : \|\beta\|_1 \le \delta\}$. Using complexity guarantees concerning duality gaps for Frank-Wolfe, we establish the following computational complexity bounds that apply simultaneously for both problems (1) and (8).

**Theorem 3.1. (Complexity of modified $\text{FS}_\varepsilon$)** *Suppose that we run the Frank-Wolfe method to solve problem (1) (with optimal objective function value $L_\delta^*$), with step-size $\bar{\alpha}_i = \frac{2}{i+2}$ . Then, after $k \ge 1$ iterations, there exists an iteration $i \in \{1, \dots, k\}$ for which:*

$$\tfrac{1}{2}\|\mathbf{y} - \mathbf{X}\beta^i\|_2^2 - L_\delta^* \le \frac{17.4\|\mathbf{X}\|_{1,2}^2 \delta^2}{k}$$

$$\|\mathbf{X}^T(\mathbf{y} - \mathbf{X}\beta^i)\|_\infty + \frac{1}{2\delta}\|\mathbf{X}\beta^i\|_2^2 - f_\delta^* \le \frac{17.4\|\mathbf{X}\|_{1,2}^2 \delta}{k}$$

$$\|\mathbf{X}^T(\mathbf{y} - \mathbf{X}\beta^i)\|_\infty \le \frac{1}{2\delta}\|\mathbf{y}\|_2^2 + \frac{17.4\|\mathbf{X}\|_{1,2}^2 \delta}{k} \ ,$$

*in addition to satisfying $\|\beta^i\|_1 \le \delta$, $\|\beta^i\|_0 \le i$. If instead we fix the number of iterations $k \ge 1$ a priori, set $\bar{\alpha}_0 := 1$ and then use an appropriately chosen constant step-size (11) for $k \ge 1$, then the $O(\frac{1}{k})$ terms in the above inequalities become $O(\frac{\log k}{k})$ terms.*

## 4 Extensions and Conclusions

Many of the results of this paper extend to the problem of boosting. Moreover, all of the methods considered in this paper fall into the broad family of dual averaging methods to solve (5) [2], which immediately yield bounds for (5). Additional references and comments can be found in the Supplementary Materials section A.

## References


[1] T. Hastie, R. Tibshirani and J. Friedman *The Elements of Statistical Learning: Prediction, Inference and Data Mining* 2nd edition, Springer Verlag, New York , 2009.

[2] Y. Nesterov. Primal-dual subgradient methods for convex problems. Mathematical Programming, 120(1) 221–259, 2009

[3] B. Polyak. *Introduction to Optimization.* Optimization Software, Inc., Publications Division, 1987.

[4] R. Tibshirani. Regression shrinkage and selection via the lasso. Journal of Royal Statistical Society B, 58 267–288, 1996.

# Fixed-Size Pegasos for Large Scale Pinball Loss SVM

Vilen Jumutc
KULeuven, ESAT-SISTA
Kasteelpark Arenberg 10,
B-3001, Leuven, Belgium
vilen.jumutc@esat.kuleuven.be

Xiaolin Huang
KULeuven, ESAT-SISTA
Kasteelpark Arenberg 10,
B-3001, Leuven, Belgium

Johan A.K. Suykens
KULeuven, ESAT-SISTA
Kasteelpark Arenberg 10,
B-3001, Leuven, Belgium

**Abstract**: Here we address the problem of learning pinball loss SVM model. We utilize modified Pegasos algorithm for large scale problems. The latter type of problems is leveraged for linearly non-separable cases via Nyström approximation and Fized-Size approach. We present an outline of the improved Pegasos algorithm and a complete learning procedure within the Fixed-Size setting.

**Keywords**: Stochastic gradient method, Pinball loss SVM, Fixed-Size approach.

## 1 Introduction

Recent research in linear Support Vector Machines (SVM) [1, 4, 8] justified the importance of the first order stochastic approaches in bringing these machine learning techniques to large scale. Computation of the full gradient sometimes might be not feasible while stochastic approximation [7] to the original optimization problem only to some degree increases the number of iterations to converge[1]. Here we consider another aspect of learning SVM models in the primal and particularly put our attention to using another loss function.

Pegasos [8] has become a widely acknowledged algorithm for learning linear SVM and has attracted research interest because of the strongly convex optimization objective and better convergence bounds. Pegasos utilizes the hinge loss which replaces the original linear constraints while making the SVM objective unconstrained. With the proper projection step Pegasos achieves a solution of accuracy $\epsilon$ in $O(\frac{R^2}{\lambda\epsilon})$ iterations where $\lambda$ is the regularization parameter and $R$ is the radius of the smallest ball containing all training samples. We would like to stress the fact that the hinge loss plays an important but not the essential role in establishing the results of Pegasos.

Here we enrich the class of loss functions applicable for Pegasos with the pinball loss [3]. We show some advantages and potential strengths of using the pinball loss within the Pegasos framework. We apply Pegasos together with the Fixed-Size approach [10, 11] to achieve better classification accuracy and to extend the method to the nonlinear case.

[1] We should note that without any stronger assumptions about our data the stochastic gradient approach delivers sublinear convergence rate compared to the linear one in the full gradient approach.

## 2 Pinball loss SVM

The sensitivity to noise or the instability to re-sampling comes from the fact that in hinge loss SVM, the distance between two sets is measured by the nearest points. Hence, one way to overcome this weak point is to change the definition of distance between two sets. For example, if we use the distance of the nearest 30% points to measure the distance between two sets, the results are less sensitive. Such distance is a kind of quantile value, which is closely related to pinball loss $L_\tau$ defined by

$$L_\tau(w;(x,y)) = \begin{cases} 1 - y\langle w,x\rangle & y\langle w,x\rangle \leq 1, \\ \tau(y\langle w,x\rangle - 1), & y\langle w,x\rangle > 1, \end{cases} \quad (1)$$

where the reasonable range of $\tau$ is $[0,1]$ as explained in [3] and the related SVM decision function is defined by $f(x) = \langle w,x\rangle$. The pinball loss $L_\tau$ has been applied for quantile regression, see, e.g. [6] [2] and [9]. Motivated by the relationship between pinball loss and quantile value, we proposed the following pinball loss SVM in [3],

$$\min_w \quad \frac{\lambda}{2}\|w\|^2 + \frac{1}{m}\sum_{(x,y)\in\mathcal{S}} L_\tau(w;(x,y)). \quad (2)$$

Hinge loss is a special case of pinball loss in Eq.(1) with $\tau = 0$. Accordingly, pinball loss SVM in Eq.(2) is an extension to hinge loss SVM. It has been shown in [3] that pinball and hinge loss SVMs have similar computational complexity and consistency property. Besides, the result of pinball loss SVM is less sensitive to noise around the boundary.

## 3 Pegasos with Pinball Loss

### 3.1 Outline of the algorithm

In Algorithm 1 we can see a major "for" loop where gradient and projection steps are taking place and a

**Algorithm 1:** Pagasos with pinball loss

**Data**: $\mathcal{S}, \lambda, \tau, T, k, \epsilon$
Select $w_1$ randomly s.t. $\|w^{(1)}\| \leq 1/\sqrt{\lambda}$
**for** $t = 1 \rightarrow T$ **do**
  Set $\eta_t = \frac{1}{\lambda t}$
  Select $\mathcal{A}_t \subseteq \mathcal{S}$, where $|\mathcal{A}_t| = k$
  $\rho = \frac{1}{|\mathcal{S}|} \sum_{(x,y)\in\mathcal{A}_t} (y - \langle w_t, x\rangle)$,
  $\mathcal{A}_t^+ = \{(x,y) \in \mathcal{A}_t : y(\langle w_t, x\rangle + \rho) < 1\}$,
  $\mathcal{A}_t^- = \{(x,y) \in \mathcal{A}_t : y(\langle w_t, x\rangle + \rho) > 1\}$,
  $w_{t+\frac{1}{2}} =$
  $w_t - \eta_t(\lambda w_t - \frac{1}{k}\left[\sum_{(x,y)\in\mathcal{A}_t^+} yx - \sum_{(x,y)\in\mathcal{A}_t^-} \tau yx\right])$
  $w_{t+1} = \min\left\{1, \frac{1/\sqrt{\lambda}}{\|w_{t+\frac{1}{2}}\|}\right\} w_{t+\frac{1}{2}}$
  **if** $\|w_{t+1} - w_t\| \leq \epsilon$ **then**
    **return** $(w_{t+1}, \frac{1}{|\mathcal{S}|} \sum_{(x,y)\in\mathcal{S}} (y - \langle w_t, x\rangle))$
  **end**
**end**
**return** $(w_{T+1}, \frac{1}{|\mathcal{S}|} \sum_{(x,y)\in\mathcal{S}} (y - \langle w_t, x\rangle))$

minor "if" condition which terminates execution if the norm of the difference of two subsequent $w$ vectors is less than $\epsilon$. In Algorithm 1 we denote the whole dataset by $\mathcal{S}$ and at each iteration select randomly $k$ samples for computation of the subgradient

$$\nabla_t = \lambda w_t - \frac{1}{|\mathcal{A}_t|}\left[\sum_{(x,y)\in\mathcal{A}_t^+} yx - \sum_{(x,y)\in\mathcal{A}_t^-} \tau yx\right] \quad (3)$$

where $\nabla_t$ additionally depends on the $\tau$ parameter of the pinball loss and $\mathcal{A}_t^-$ stands for the subset of $\mathcal{A}_t$ where $y\langle w, x\rangle > 1$ and $\mathcal{A}_t^+$ is the reciprocal subset where $y\langle w, x\rangle < 1$. For additional details and supplementary analysis we refer to our paper [5]. Another important issue is related to the computation of the bias term. We should emphasize that the bias term $\rho$ is not part of our instantaneous optimization objective and we perform computation of it just to return convenient and ubiquitous representation of the SVM decision function by $\hat{y} = \text{sign}(\langle w, x\rangle + \rho)$ where $\rho$ is returned in the Line 10 and 13 of Algorithm 1 together with $w$ vector.

### 3.2 Complete procedure

In Algorithm 2 the "PegasosPBL" function stands for the shortcut of Algorithm 1. The "ComputeNystromApprox" function denotes the Fixed-Size part where we first compute $m \times m$ RBF kernel matrix[2] from the data points found by the maximization of Renyi entropy in "FindActiveSet" function and then we apply Nyström approximation

$$\hat{\Phi}_i(x) = \frac{1}{\sqrt{\lambda_{i,m}}} \sum_{t=1}^{m} u_{ti,m} k(x_t, x), \quad (4)$$

[2] we assume $m \ll |\mathcal{S}|$

**Algorithm 2:** Fized-Size Pegasos with pinball loss

**input** : training data $\mathcal{S}$ with $|\mathcal{S}| = n$, labeling $Y$, parameters $\lambda, \tau, T, k, \epsilon, m$
**output**: mapping $\hat{\Phi}(x), \forall x \in \mathcal{S}$, SVM model given by $w$ and $\rho$
**begin**
  $\mathcal{S}_r \leftarrow$ `FindActiveSet`$(\mathcal{S}, m)$;
  $\hat{\Phi}(x) \leftarrow$ `ComputeNystromApprox`$(\mathcal{S}_r)$;
  $X \leftarrow [\hat{\Phi}(x_1)^T, \ldots, \hat{\Phi}(x_n)^T]$;
  $[w, \rho] \leftarrow$ `PegasosPBL`$(X, Y, \lambda, \tau, T, k, \epsilon)$;
**end**

where $\lambda_{i,m}$ and $u_{i,m}$ denote the $i$-th eigenvalue and the $i$-th eigenvector of the RBF kernel matrix $K$ to derive our approximate feature map $\hat{\Phi}(x)$. Finally we stack our explicit feature vectors in matrix $X$ and proceed to the function "PegasosPBL".


### Acknowledgments

This work was supported by Research Council KUL, ERC AdG A-DATADRIVE-B, GOA/10/09MaNet, CoE EF/05/006, FWO G.0588.09, G.0377.12, SBO POM, IUAP P6/04 DYSCO, COST intelliCIS.


## References


[1] O. Chapelle. Training a support vector machine in the primal. *Neural Computation*, 19:1155–1178, 2007.

[2] A. Christmann and I. Steinwart. How SVMs can estimate quantiles and the median. In *NIPS*, pages 305–312, 2007.

[3] X. Huang, L. Shi, and J. A. K. Suykens. Support vector machine classifier with pinball loss. Technical Report KUL-12-162, Katholieke Universiteit Leuven, Kasteelpark Arenberg 10, Leuven, 2012. Available: `http://homes.esat.kuleuven.be/~sistawww/cgi-bin/newsearch.pl?Name=Huang+X`

[4] T. Joachims. Training linear SVMs in linear time. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '06, pages 217–226, New York, NY, USA, 2006. ACM.

[5] V. Jumutc, X. Huang, and J. A. K. Suykens. Fixed-size pegasos for hinge and pinball loss SVM. Technical Report KUL-13-31, Katholieke Universiteit Leuven, Kasteelpark Arenberg 10, Leuven, 2013.

[6] R. Koenker. *Quantile Regression*. Econometric Society Monographs. Cambridge University Press, 2005.

[7] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM J. on Optimization*, 19(4):1574–1609, January 2009.

[8] S. Shalev-Shwartz, Y. Singer and N. Srebro. Pegasos: Primal Estimated sub-GrAdient SOlver for SVM. In *Proceedings of the 24th international conference on Machine learning*, ICML '07, pages 807–814, New York, NY, USA, 2007.

[9] I. Steinwart and A. Christmann. Estimating conditional quantiles with the help of the pinball loss. *Bernoulli*, 17(1):211–225, 2011.

[10] 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.

[11] C. Williams and M. Seeger. Using the Nyström method to speed up kernel machines. In *Advances in Neural Information Processing Systems 13*, pages 682–688. MIT Press, 2001.

# Output Kernel Learning Methods

Francesco Dinuzzo
MPI for Intelligent Systems
Tübingen, Germany
fdinuzzo@tuebingen.mpg.de

Cheng Soon Ong
NICTA,
Melbourne, Australia
chengsoon.ong@unimelb.edu.au

Kenji Fukumizu
Institute of Statistical Mathematics,
Tachikawa, Tokyo, Japan
fukumizu@ism.ac.jp

**Abstract**: A rather flexible approach to multi-task learning consists in solving a regularization problem where a suitable kernel is used to model joint relationships between both inputs and tasks. Since specifying an appropriate multi-task kernel in advance is not always possible, estimating one from the data is often desirable. Herein, we overview a class of techniques for learning a multi-task kernel that can be decomposed as the product of a kernel on the inputs and one on the task indices. The kernel on the task indices (output kernel) is optimized simultaneously with the predictive function by solving a joint two-level regularization problem.

**Keywords**: regularization, multi-task learning, kernel learning

## 1 Learning Multi-Task Kernels

Predictive performances of kernel-based regularization methods are highly influenced by the choice of the kernel function. Such influence is especially evident in the case of multi-task learning where, in addition to specifying input similarities, it is crucial to correctly model inter-task relationships. Designing the kernel allows to incorporate domain knowledge by properly constraining the function class over which the solution is searched. Unfortunately, in many problems the available knowledge is not sufficient to uniquely determine a good kernel in advance, making it highly desirable to have data-driven automatic selection tools. This need has motivated a fruitful research stream which has led to the development of a variety of techniques for learning the kernel.

For a broad class of multi-task (or multi-output) learning problems, a kernel can be used to specify the joint relationships between inputs and tasks [1]. Generally, it is necessary to specify similarities of the form $K((x_1, i), (x_2, j))$ for every pair of input data $(x_1, x_2)$ and every pair of task indices $(i, j)$. However, a very common way to simplify modeling is to utilize a multiplicative decomposition of the form

$$K((x_1, i), (x_2, j)) = K_X(x_1, x_2)K_Y(i, j),$$

where the *input kernel* $K_X$ is decoupled from the *output kernel* $K_Y$. The same structure can be equivalently represented in terms of a matrix-valued kernel

$$H(x_1, x_2) = K_X(x_1, x_2) \cdot \mathbf{L}, \tag{1}$$

where $\mathbf{L}$ is a symmetric and positive semidefinite matrix with entries $\mathbf{L}_{ij} = K_Y(i, j)$.

Even after imposing such simplified model, specifying the inter-task similarities in advance may still be impractical. Indeed, it is often the case that multiple learning tasks are known to be related, but no precise information about the structure or the intensity of such relationships is available. Simply fixing $\mathbf{L}$ to the identity is clearly suboptimal since it amounts to share no information between the tasks. On the other hand, wrongly specifying the entries may lead to a severe performance degradation. It is therefore clear that, whenever the task relationships are subject to uncertainty, learning them from the data is the only meaningful way to proceed.

The most widely developed approach to automatic kernel selection, known as Multiple Kernel Learning (MKL), consists in learning a conic combination of basis kernels of the form

$$K = \sum_{k=1}^{N} d_k K_k, \qquad d_k \geq 0.$$

Appealing properties of MKL methods include the ability to perform selection of a subset of kernels via sparsity, and tractability of the associated optimization

problem, typically (re)formulated as a convex program. Apparently, the MKL approach can be also used to learn a multi-task kernel of the form

$$K((x_1,i),(x_2,j)) = \sum_{k=1}^{N} d_k K_X^k(x_1,x_2) K_Y^k(i,j),$$

that includes the possibility of optimizing the matrix $\mathbf{L}$ in (1) as a conic combination of basis matrices. In principle, proper complexity control allows to combine an arbitrarily large, even infinite [2], number of kernels. However, computational and memory constraints force the user to specify a relatively small dictionary of basis kernels to be combined, which again calls for a certain amount of domain knowledge.

## 2 Output Kernel Learning

A more direct approach to synthesize the output kernel from the data consists in solving a two-level regularization problem of the form

$$\min_{\mathbf{L}\in\mathbb{S}_+} \left[ \min_{f\in\mathcal{H}_\mathbf{L}} \left( \sum_{i=1}^{\ell}\sum_{j=1}^{m} V(y_{ij}, f_j(x_{ij})) + \|f\|^2_{\mathcal{H}_\mathbf{L}} + \Omega(\mathbf{L}) \right) \right],$$

where $V$ is a suitable loss function, $\mathcal{H}_\mathbf{L}$ is the Reproducing Kernel Hilbert Space of vector-valued functions associated with the reproducing kernel (1), $\Omega$ is a suitable matrix regularizer, and $\mathbb{S}_+$ is the cone of symmetric and positive semidefinite matrices. We call such an approach Output Kernel Learning (OKL).

A technique of this kind was introduced in [3] for the case where $L$ is a square loss function, $\Omega$ is the squared Frobenius norm, and the input data $x_{ij}$ are the same for all the output components $f_j$. Such special structure of the objective functional allows to develop an effective block coordinate descent strategy where each step involves the solution of a Sylvester linear matrix equation. Regularizing the output kernel with a squared Frobenius norm leads to a simple and effective computational scheme. However, we may want to encourage different types of relationship structures in the output space. Along this line, [4] introduces low-rank OKL, a method to discover relevant low dimensional subspaces of the output space by learning a low-rank kernel matrix. This is achieved by regularizing the output kernel with a combination of the trace and a rank indicator function, namely

$$\Omega(\mathbf{L}) = \mathrm{tr}(\mathbf{L}) + I(\mathrm{rank}(\mathbf{L}) \leq p).$$

For $p = m$, the hard-rank constraint disappears and $\Omega$ reduces to the trace norm which, as it is well known, encourages low-rank solutions. Setting $p < m$ gives up convexity of the regularizer but, on the other hand, allows to set a hard bound on the rank of the output kernel, which can be useful for both computational and interpretative reasons. Low-rank OKL enjoys interesting properties and interpretations. Just as sparse MKL with a square loss can be seen as a nonlinear generalization of (grouped) Lasso, low-rank OKL is a natural kernel-based generalization of reduced-rank regression, a popular multivariate technique in statistics.

For problems where the inputs $x_{ij}$ are the same for all the tasks, optimization for low-rank OKL can be performed by means of a rather effective procedure that iteratively computes eigendecompositions. Importantly, the size of the involved matrices can be controlled by selecting the parameter $p$. Unfortunately, more general multi-task learning problems where each task is sampled in correspondence with different inputs require completely different methods. If a square loss is adopted, it turns out that an effective strategy to approach these problems consists in iteratively apply inexact Preconditioned Conjugate Gradient (PCG) solvers [5] to suitable linear operator equations that arise from the optimality conditions.

## 3 Concluding remarks and future directions

Learning output kernels via regularization is an effective way to solve multi-task learning problems where the relationships between the tasks are highly uncertain. The OKL framework that we have sketched in the previous section is rather general and can be developed in various directions. Effective optimization techniques for more general (non-quadratic) loss functions are still lacking and the use of a variety of matrix penalties for the output kernel matrix is yet to be explored.

## References


[1] T. Evgeniou, C. A. Micchelli, and M. Pontil. Learning multiple tasks with kernel methods. *Journal of Machine Learning Research*, 6:615–637, 2005.

[2] A. Argyriou, C. A. Micchelli, and M. Pontil. Learning convex combinations of continuously parameterized basic kernels. In Peter Auer and Ron Meir, editors, *Learning Theory*, volume 3559 of *Lecture Notes in Computer Science*, pages 338–352. Springer Berlin / Heidelberg, 2005.

[3] F. Dinuzzo, C. S. Ong, P. Gehler, and G. Pillonetto. Learning output kernels with block coordinate descent. In *Proceedings of the 28th Annual International Conference on Machine Learning*, Bellevue, WA, USA, 2011.

[4] F. Dinuzzo and K. Fukumizu. Learning low-rank output kernels. *Journal of Machine Learning Research - Proceedings Track*, 20:181–196, 2011.

[5] F. Dinuzzo. Learning output kernels for multi-task problems. *Neurocomputing*, (to appear), 2013.

# Deep Support Vector Machines for Regression Problems

M.A. Wiering, M. Schutten, A. Millea, A. Meijster, and L.R.B. Schomaker
Institute of Artificial Intelligence and Cognitive Engineering
University of Groningen, the Netherlands
Contact e-mail: m.a.wiering@rug.nl

**Abstract**: In this paper we describe a novel extension of the support vector machine, called the deep support vector machine (DSVM). The original SVM has a single layer with kernel functions and is therefore a shallow model. The DSVM can use an arbitrary number of layers, in which lower-level layers contain support vector machines that learn to extract relevant features from the input patterns or from the extracted features of one layer below. The highest level SVM performs the actual prediction using the highest-level extracted features as inputs. The system is trained by a simple gradient ascent learning rule on a min-max formulation of the optimization problem. A two-layer DSVM is compared to the regular SVM on ten regression datasets and the results show that the DSVM outperforms the SVM.

**Keywords**: Support Vector Machines, Kernel Learning, Deep Architectures

## 1 Introduction

Machine learning algorithms are very useful for regression and classification problems. These algorithms learn to extract a predictive model from a dataset of examples containing input vectors and target outputs. Among all machine learning algorithms, one of the most popular methods is the SVM. SVMs have been used for many engineering applications such as object recognition, document classification, and different applications in bio-informatics, medicine and chemistry.

**Limitations of the SVM.** There are two important limitations of the standard SVM. The first one is that the standard SVM only has a single adjustable layer of model parameters. Instead of using such "shallow models", deep architectures are a promising alternative [4]. Furthermore, SVMs use a-priori chosen kernel functions to compute similarities between input vectors. A problem is that using the best kernel function is important, but kernel functions are not very flexible.

**Related Work.** Currently there is a lot of research in multi-kernel learning (MKL) [1, 5]. In MKL, different kernels are combined in a linear or non-linear way to create more powerful similarity functions for comparing input vectors. However, often only few parameters are adapted in the (non-linear) combination functions. In [2], another framework for two-layer kernel machines is described, but no experiments were performed in which both layers used non-linear kernels.

**Contributions.** We propose the deep SVM (DSVM), a novel algorithm that uses SVMs to learn to extract higher-level features from the input vectors, after which these features are given to the main SVM to do the actual prediction. The whole system is trained with simple gradient ascent and descent learning algorithms on the dual objective of the main SVM. The main SVM learns to maximize this objective, while the feature-layer SVMs learn to minimize it. Instead of adapting few kernel weights, we use large DSVM architectures, sometimes consisting of a hundred SVMs in the first layer. Still, the complexity of our DSVM scales only linearly with the number of SVMs compared to the standard SVM. Furthermore, the strong regularization power of the main SVM prevents overfitting.

## 2 The Deep Support Vector Machine

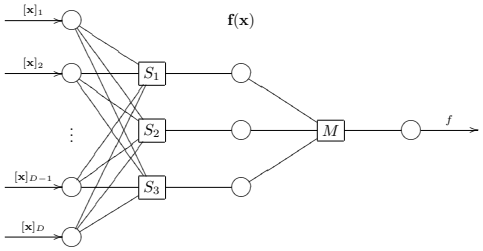



**Fig. 1:** Architecture of a two-layer DSVM. In this example, the feature layer consists of three SVMs $S_a$.

We use regression datasets: $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_\ell, y_\ell)\}$, where $\mathbf{x}_i$ are input vectors and $y_i$ are the target outputs. The architecture of a two-layer DSVM is shown in Figure 1. First, it contains an input layer of $D$ inputs. Then, there are a total of $d$ pseudo-randomly initialized SVMs $S_a$, each one learning to extract one feature $\mathbf{f}(\mathbf{x})_a$ from an input pattern $\mathbf{x}$. Finally, there is the main support vector machine $M$ that approximates the target function using the extracted feature vector as

input. For computing the feature-layer representation $\mathbf{f}(\mathbf{x})$ of input vector $\mathbf{x}$, we use:

$$\mathbf{f}(\mathbf{x})_a = \sum_{i=1}^{\ell} (\alpha_i^*(a) - \alpha_i(a)) K(\mathbf{x}_i, \mathbf{x}) + b_a,$$

which iteratively computes each element $\mathbf{f}(\mathbf{x})_a$. In this equation, $\alpha_i^*(a)$ and $\alpha_i(a)$ are SVM coefficients for SVM $S_a$, $b_a$ is its bias, and $K(\cdot,\cdot)$ is a kernel function. For computing the output of the whole system, we use:

$$g(\mathbf{f}(\mathbf{x})) = \sum_{i=1}^{\ell} (\alpha_i^* - \alpha_i) K(\mathbf{f}(\mathbf{x}_i), \mathbf{f}(\mathbf{x})) + b.$$

**Learning Algorithm.** The learning algorithm adjusts the SVM coefficients of all SVMs through a min-max formulation of the dual objective $W$ of the main SVM:

$$\min_{\mathbf{f}(\mathbf{x})} \max_{\boldsymbol{\alpha},\boldsymbol{\alpha}^*} W(\mathbf{f}(\mathbf{x}), \boldsymbol{\alpha}^{(*)}) = -\varepsilon \sum_{i=1}^{\ell} (\alpha_i^* + \alpha_i) + \sum_{i=1}^{\ell} (\alpha_i^* - \alpha_i) y_i$$
$$-\frac{1}{2} \sum_{i,j=1}^{\ell} (\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j) K(\mathbf{f}(\mathbf{x}_i), \mathbf{f}(\mathbf{x}_j))$$

We have developed a simple gradient ascent algorithm to train the SVMs. The method adapts the SVM coefficients $\boldsymbol{\alpha}^{(*)}$ (standing for all $\alpha_i^*$ and $\alpha_i$) toward a (local) maximum of $W$, where $\lambda$ is the learning rate: $\alpha_i^{(*)} \leftarrow \alpha_i^{(*)} + \lambda \cdot \partial W / \partial \alpha_i^{(*)}$. The resulting gradient ascent learning rule for $\alpha_i$ is:

$$\alpha_i = \alpha_i + \lambda(-\epsilon - y_i + \sum_j (\alpha_j^* - \alpha_j) K(\mathbf{f}(\mathbf{x}_i), \mathbf{f}(\mathbf{x}_j)))$$

We use radial basis function (RBF) kernels in both layers of a two-layered DSVM. Results with other kernels were worse. For the main SVM:

$$K(\mathbf{f}(\mathbf{x}_i), \mathbf{f}(\mathbf{x})) = \exp(-\sum_a \frac{(\mathbf{f}(\mathbf{x}_i)_a - \mathbf{f}(\mathbf{x})_a)^2}{\sigma_m})$$

The system constructs a new dataset for each feature-layer SVM $S_a$ with a backpropagation-like technique for making examples: $(\mathbf{x}_i, \mathbf{f}(\mathbf{x}_i)_a - \mu \cdot \delta W / \delta \mathbf{f}(\mathbf{x}_i)_a)$, where $\mu$ is some learning rate, and $\delta W / \delta \mathbf{f}(\mathbf{x}_i)_a$ is given by:

$$\frac{\delta W}{\delta \mathbf{f}(\mathbf{x}_i)_a} = (\alpha_i^* - \alpha_i) \sum_{j=1}^{\ell} (\alpha_j^* - \alpha_j) \frac{\mathbf{f}(\mathbf{x}_i)_a - \mathbf{f}(\mathbf{x}_j)_a}{\sigma_m} \cdot K(\mathbf{f}(\mathbf{x}_i), \mathbf{f}(\mathbf{x}_j))$$

The feature extracting SVMs are pseudo-randomly initialized and then alternated training of the main SVM and feature layer SVMs is executed a number of epochs. The bias values are computed from the average errors.

## 3 Experimental Results

We experimented with 10 regression datasets to compare the DSVM to an SVM, both using RBF kernels. Both methods are trained with our simple gradient ascent learning rule, adapted to also consider penalties, e.g. for obeying the bias constraint. The first 8 datasets are described in [3] and the other 2 datasets are taken from the UCI repository. The number of examples per dataset ranges from 43 to 1049, and the number of features is between 2 and 13. The datasets are split into 90% trainingdata and 10% testingdata. For optimizing the learning parameters we have used particle swarm optimization. Finally, we used 1000 or 4000 times cross-validation with the best found parameters to compute the mean squared error and its standard error.

| Dataset | SVM results | DSVM results |
|---|---|---|
| Baseball | 0.02413 ± 0.00011 | **0.02294** ± 0.00010 |
| Boston H. | 0.006838 ± 0.000095 | **0.006381** ± 0.000090 |
| Concrete . | 0.00706 ± 0.00007 | **0.00621** ± 0.00005 |
| Electrical | 0.00638 ± 0.00007 | 0.00641 ± 0.00007 |
| Diabetes | 0.02719 ± 0.00026 | **0.02327** ± 0.00022 |
| Machine-CPU | 0.00805 ± 0.00018 | **0.00638 ± 0.00012** |
| Mortgage | 0.000080 ± 0.000001 | 0.000080 ± 0.000001 |
| Stock | 0.000862 ± 0.000006 | **0.000757 ± 0.000005** |
| Auto-MPG | 6.852 ± 0.091 | 6.715 ± 0.092 |
| Housing | **8.71 ± 0.14** | 9.30 ± 0.15 |

**Tab. 1:** The mean squared errors and standard errors of the SVM and the two-layer DSVM on 10 regression datasets.

Table 1 shows the results. The results of the DSVM are significantly better for 6 datasets ($p < 0.001$) and worse on one. From the results we can conclude that the DSVM is a powerful novel machine learning algorithm. More research, such as adding more layers and implementing more powerful techniques to scale up to big datasets, can be done to discover its full potential.

## References


[1] F.R. Bach, G.R.G. Lanckriet, and M.I. Jordan. Multiple kernel learning, conic duality, and the SMO algorithm. In *Proceedings of the twenty-first international conference on Machine learning*, ICML '04, pages 6–15, 2004.

[2] Francesco Dinuzzo. Kernel machines with two layers and multiple kernel learning. *CoRR*, 2010.

[3] M. Graczyk, T. Lasota, Z. Telec, and B. Trawinski. Nonparametric statistical analysis of machine learning algorithms for regression problems. In *Knowledge-Based and Intelligent Information and Engineering Systems*, pages 111–120. 2010.

[4] G.E. Hinton and R.R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313:504–507, 2006.

[5] S. Sonnenburg, G. Rätsch, C. Schäfer, and B. Schölkopf. Large scale multiple kernel learning. *Journal of Machine Learning Research*, 7:1531–1565, 2006.

# Subspace Learning and Empirical Operator Estimation

Alessandro Rudi
Istituto Italiano di Tecnologia
alessandro.rudi@iit.it

Guillermo D. Canas
MIT-IIT
guilledc@mit.edu

Lorenzo Rosasco
Università di Genova & MIT-IIT
lrosasco@mit.edu

**Abstract**: This work deals with the problem of linear subspace estimation in a general, Hilbert space setting. We provide bounds that are considerably sharper than existing ones, under equal assumptions. These bounds are also competitive with bounds that are allowed to make strong, further assumptions (on the fourth order moments), even when we do not. Finally, we generalize these results to a family of metrics, allowing for a more general definition of performance.

**Keywords**: Subspace Learning, PCA, Kernel-PCA

## 1 Introduction

Estimating the smallest linear space supporting data drawn from an unknown distribution is a classical problem in machine learning and statistics, with several stablished algorithms addressing it, most notably PCA and kernel PCA [2, 3]. Thus knowledge of the speed of convergence of these estimators with respect to the sample size, and the algorithms' parameters is of considerable practical importance.

We use tools from linear operator theory to arrive at novel learning rates for the subspace estimation problem. These rates are significantly sharper than existing ones, under typical assumptions on the eigenvalue decay rate of the covariance. Furthermore, they cover a wider range of performance metrics.

**Problem statement**. Given a measure $\rho$ with support $M$ in the unit ball of a separable Hilbert space $\mathcal{H}$, we consider the problem of estimating, from $n$ i.i.d. samples $X_n = \{x_i\}_{i=1}^n$, the smallest linear subspace $S_\rho$ that includes $M$, that is, $S_\rho := \overline{\text{span}(M)}$.

The quality of an an estimated subspace $\hat{S}$, for a given metric (pseudo-metric) $d$, is characterized in terms of bounds of the form

$$\mathbb{P}\left[d(S_\rho, \hat{S}) \leq \varepsilon(\delta, n)\right] \geq 1 - \delta, \quad 0 < \delta < 1. \qquad (1)$$

Common choices of $\hat{S}$ are the empirical estimate $\hat{S}_n := \text{span}(X_n)$, and the $k$-truncated (kernel) PCA estimate $\hat{S}_n^k$ (where $\hat{S}_n^n = \hat{S}_n$).

**Performance criteria**. A natural performance criteria is the so-called *reconstruction pseudometric*

$$d_R(S_\rho, \hat{S}) := \mathbb{E}_{x\sim\rho}\|P_{S_\rho}(x) - P_{\hat{S}}(x)\|_{\mathcal{H}}^2$$

where $P_V$ is the metric projection onto a subspace $V$. Another important criterion is the *gap distance* $d_G(S_\rho, \hat{S}) := \|P_{S_\rho} - P_{\hat{S}}\|_\infty$, which measures, inside the unit ball, the maximum distance to $\hat{S}$ over points in $S_\rho$.

By letting $C_\rho := \mathbb{E}_{x\sim\rho} x \otimes x$ be the (uncentered) covariance operator associated to $\rho$, and defining

$$d_{\alpha,p}(S_\rho, \hat{S}) := \|(P_{S_\rho} - P_{\hat{S}})C_\rho^\alpha\|_p \qquad (2)$$

we notice that it is both $d_R(S_\rho, \cdot) = d_{1/2,2}(S_\rho, \cdot)^2$, and $d_G = d_{0,\infty}$, and therefore the two metrics can be analyzed using common tools.

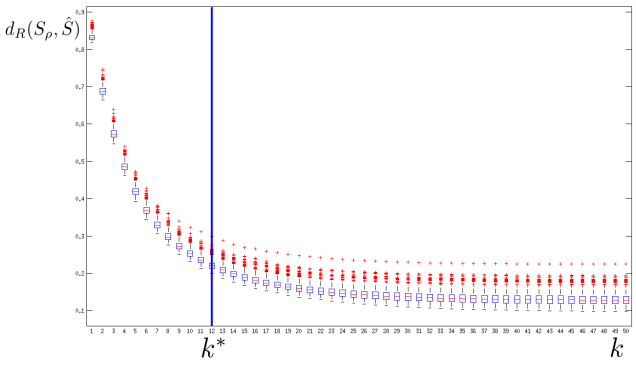



**Figure 1:** Expected distance from a random sample to the empirical $k$-truncated kernel-PCA -subspace estimate, as a function of $k$ ($n = 1000$, 1000 trials shown in a boxplot). Our predicted plateau threshold $k^*$ (Cor. 2.2) is a good estimate of the value $k$ past which the distance stabilizes.

### 1.1 Example

Consider a simple one-dimensional uniform distribution embedded into a reproducing-kernel Hilbert space $\mathcal{H}$ (using the exponential of the $\ell_1$ distance as kernel). Figure 1 is a box plot of $d_R(S_\rho, \hat{S}_n^k)$, where $\hat{S}_n^k$ is the $k$-truncated kernel-PCA estimate, with $n = 1000$ and varying $k$. Note that, while $d_R$ is computed analytically in this example, and $S_\rho$ is fixed, the estimate $\hat{S}_n^k$ is a random variable, and hence the variability in the graph. The graph is highly concentrated around a curve with a steep intial drop, until reaching some sufficiently high $k$, past which the reconstruction (pseudo) distance becomes stable, and does not vanish. In our experiments, this behavior is typical for the reconstruction distance

and high-dimensional problems.

Notice that our bound for this case (Cor. 2.3) similarly predicts a steep performance drop until a value $k = k^*$ (indicated in the figure by the vertical blue line), and a plateau afterwards.

## 2 Learning Rates

Our main technical contribution is a bound of the form of Eq. (1). We begin by stating it in the most general form in Th. 2.1, which bounds the general distance $d_{\alpha,p}$ given a known covariance $C_\rho$.

**Theorem 2.1.** *Let $\{x_i\}_{i=1}^n$ be drawn i.i.d. according to a probability measure $\rho$ supported on the unit ball of a separable Hilbert space $\mathcal{H}$, with covariance $C_\rho$. Assuming $n > 3$, $0 < \delta < 1$, $0 \leq \alpha \leq \frac{1}{2}$ and $k \geq 1$, it holds*

$$\mathbb{P}\left[d_{\alpha,p}(S_\rho, \hat{S}_n^k) \leq 4t^\alpha \left\|C_\rho^\alpha (C_\rho + t)^{-\alpha}\right\|_p\right] \geq 1 - \delta$$

*where $t = \max\{\sigma_k, \frac{9}{n}\log\frac{n}{\delta}\}$, and $\sigma_k$ is the $k$-th top eigenvalue of $C_\rho$.*

We say that $C_\rho$ has *eigenvalue decay rate of order $r$* if there are a constants $q, Q > 0$ such that $qj^{-r} \leq \sigma_j \leq Qj^{-r}$, where $\sigma_j$ are the (decreasingly ordered) eigenvalues of $C_\rho$, and $r > 1$. Such knowledge can be incorporated into Th. 2.1 to obtain explicit learning rates, as follows.

**Corollary 2.2** (Polynomial eigenvalue decay)**.** *Let $C_\rho$ have eigenvalue decay rate of order $r$. Under the assumptions of Th. 2.1, it is, with probability $1 - \delta$*

$$d_{\alpha,p}(S_\rho, \hat{S}_n^k) \leq Q' \min\{k, k^*\}^{-r\alpha + \frac{1}{p}}$$

*where $k^* = \left(\frac{qn}{9\log(n\delta^{-1})}\right)^{\frac{1}{r}}$, and $Q'$ is a constant whose value is omitted here for brevity.*

Finally, by particularizing Cor. 2.2 to the reconstruction distance $d_R$, and choosing the predicted optimal $k = k^*$, we obtain the following result.

**Corollary 2.3** (Reconstruction distance)**.** *Letting $C_\rho$ have eigenvalue decay rate of order $r$, and $k^*$ be as in Cor. 2.2, it holds with probability $1 - \delta$:*

$$d_R(S_\rho, \hat{S}_n^{k^*}) = O\left(\left(\frac{\log n}{n}\right)^c\right)$$

*where $c = 1 - 1/r$.*

Note that, as is, Th. 2.1 does not provide a useful bound for the gap-metric $d_{0,\infty}$. We obtain bounds for this metric independently, but they are omitted here in the interest of conciseness.

## 3 Discussion

Figure 2 shows a comparison of our learning rates with existing rates in the literature [1, 4]. The plot shows the polynomial decay rate $c$ of $d_R(S_\rho, \hat{S}_n^k) = O(n^{-c})$, as a function of the eigenvalue decay rate $r$ of the covariance $C_\rho$, computed at the best value $k^*$ (which minimizes the bound).

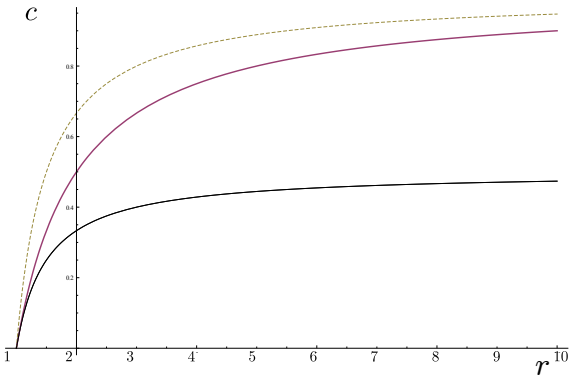



**Figure 2:** Known upper bounds for the polynomial decay rate $c$ (for the best choice of $k$), for the expected distance from a random sample to the empirical $k$-truncated kernel-PCA estimate, as a function of the covariance eigenvalue decay rate (higher is better). Our bound (purple line), consistently outperforms previous ones ([4] black line). The top (dashed) line, has significantly stronger assumptions, and is only included for completeness.

The rate exponent $c$, under a polynomial eigenvalue decay assumption for $C_\rho$, is $c = \frac{s(r-1)}{r-s+sr}$ for [1] and $c = \frac{r-1}{2r-1}$ for [4], where $s$ is related to the fourth moment. Note that, among the two (purple and black) that operate under the same assumptions, ours (purple line) is the best by a wide margin. The top, best performing, dashed line [1] is obtained for the best possible fourth-order moment $s = 2r$, and therefore it is not a fair comparison. However, it is worth noting that our bounds perform almost as well as the most restrictive one, even when we do not include any fourth-order moment constraints.

## References

[1] G. Blanchard, O. Bousquet, and L. Zwald. Statistical properties of kernel principal component analysis. *Machine Learning*, 66(2):259–294, 2007.

[2] I. Jolliffe. *Principal component analysis*. Wiley Online Library, 2005.

[3] B. Schölkopf, A. Smola, and K.R. Müller. Kernel principal component analysis. *Artificial Neural Networks—ICANN'97*, pages 583–588, 1997.

[4] J. Shawe-Taylor, C. K. Williams, N. Cristianini, and J. Kandola. On the eigenspectrum of the gram matrix and the generalization error of kernel-pca. *Information Theory, IEEE Transactions on*, 51(7):2510–2522, 2005.

# Kernel based identification of systems with multiple outputs using nuclear norm regularization

Tillmann Falck, Bart De Moor, Johan A. K. Suykens
SCD/SISTA, ESAT, KU Leuven
{tillmann.falck, bart.demoor, johan.suykens}@esat.kuleuven.be

**Abstract**: This contribution introduces a novel identification scheme for nonlinear systems with multiple outputs based on nuclear norm regularization. A kernel based formulation is derived in a primal-dual setting. Then the model representation in terms of the kernel function is established.

**Keywords**: kernel methods, system identification, nuclear norm, primal-dual framework

## 1 Introduction

Kernel based methods have been proved to be a powerful technique for nonlinear system identification [6]. The established approach to handle systems with multiple outputs is to estimate independent models for each output. This approach is suboptimal however. Consider the scenario described in [1] which analyzes time series acquired from the Belgian power distribution network. The data set contains measurements from several hundred substations and the referenced work identified 7 profiles which can be used to explain the measurements of all substations. This illustrates a large amount of redundancy contained in the data which cannot be captured by estimating individual models for each output. To utilize such relationships among several variables for improved model performance, this paper proposes the use of nuclear norm regularization [4].

The combination of a kernel based model and nuclear norm regularization is still challenging. A representer theorem has been proved in [2] for a class of matrix regularizers including the nuclear norm. This contribution however adopts a novel primal-dual approach. The primal-dual formulation makes it straightforward to include prior information in terms of additional constraints and in general is very flexible as demonstrated in [5].

## 2 Problem statement

The parametric model formulation is a direct extension of support vector models in a primal-dual setting. For a pair of inputs and outputs $(\boldsymbol{x}_t, \boldsymbol{y}_t)$, the model equation is given by

$$y_t^{(i)} = \boldsymbol{w}_i^T \boldsymbol{\varphi}(\boldsymbol{x}_t) + b_i \tag{1}$$

for $i = 1, \ldots, M$ and $\boldsymbol{y}_t = [y_t^{(1)}, \ldots, y_t^{(M)}]^T$. Here $\boldsymbol{w}_i \in \mathbb{R}^{n_h}$ denotes the parameter vector for the $i$-th output and $b_i$ is the corresponding intercept. The nonlinear map $\boldsymbol{\varphi} : \mathbb{R}^D \to \mathbb{R}^{n_h}$ is referred to as feature map and often only implicitly defined by the relation $K(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{\varphi}(\boldsymbol{x})^T \boldsymbol{\varphi}(\boldsymbol{y})$. A popular choice for the positive semidefinite kernel function $K$ is the Gaussian RBF kernel $K_{\text{RBF}}(\boldsymbol{x}, \boldsymbol{y}) = \exp(-\|\boldsymbol{x} - \boldsymbol{y}\|_2^2/\sigma^2)$.

Given data $\{(\boldsymbol{x}_t, \boldsymbol{y}_t)\}_{t=1}^N$ a model of the form specified by (1) can be estimated by solving the following convex problem,

$$\begin{aligned} \min_{\boldsymbol{W}, \boldsymbol{b}, \boldsymbol{e}_t} \quad & \eta \|\boldsymbol{W}\|_* + \frac{1}{2} \sum_{t=1}^N \boldsymbol{e}_t^T \boldsymbol{e}_t \\ \text{subject to} \quad & \boldsymbol{y}_t = \boldsymbol{W}^T \boldsymbol{\varphi}(\boldsymbol{x}_t) + \boldsymbol{b} + \boldsymbol{e}_t, \ t \in \mathbb{S}, \end{aligned} \tag{2}$$

where $\mathbb{S} = \{1, \ldots, N\}$. Note the use of the nuclear norm $\|\cdot\|_*$ as regularization term. This imposes a connection between the different output variables as motivated in the introduction. However the deviation from the classical quadratic regularization term $\operatorname{vec}(\boldsymbol{W})^T \operatorname{vec}(\boldsymbol{W})$ requires a new solution strategy to obtain a kernel based model, which is outlined in the next section.

## 3 Kernel based formulation

### 3.1 Dual optimization problem

In kernel based models the parametric problem (2) can often not be solved directly as the feature map $\boldsymbol{\varphi}$ is not explicitly known. One popular way of obtaining a tractable problem is deriving the Lagrange dual. To handle the nondifferentiable objective one can reformulate the nuclear norm by using the definition of the dual norm

$$\|\boldsymbol{W}\|_* = \max_{\|\boldsymbol{C}\|_2 \leq 1} \operatorname{tr}(\boldsymbol{C}^T \boldsymbol{W}) \tag{3}$$

where $\|\cdot\|_2$ is the spectral norm. This allows the derivation of the following lemma.

**Lemma 1.** *The solution to* (2) *is equivalent to the solution of its Lagrange dual*

$$\begin{aligned} \max_{\boldsymbol{A} \in \mathbb{R}^{M \times N}} \quad & \operatorname{tr}(\boldsymbol{A}^T \boldsymbol{Y}) - \frac{1}{2} \operatorname{tr}(\boldsymbol{A}^T \boldsymbol{A}) \\ \text{subject to} \quad & \boldsymbol{A} \boldsymbol{1}_N = \boldsymbol{0}_M, \quad \boldsymbol{A} \boldsymbol{\Omega} \boldsymbol{A}^T \preceq \eta^2 \boldsymbol{I}_M \end{aligned} \tag{4}$$

*with* $\boldsymbol{Y} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N] \in \mathbb{R}^{M \times N}$, $\boldsymbol{1}_N \in \mathbb{R}^N$ *a vector of all ones,* $\boldsymbol{0}_M \in \mathbb{R}^M$ *a vector of all zeros and* $\boldsymbol{I}_M$ *the identity matrix of size* $M$*. The elements of the Gram matrix* $\boldsymbol{\Omega}$ *are computed according to* $\Omega_{ij} = \boldsymbol{\varphi}(\boldsymbol{x}_i)^T\boldsymbol{\varphi}(\boldsymbol{x}_j) = K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ *for* $i, j = 1, \ldots, N$*.*

The proof of this lemma can be found in Chapter 6 of [3]. Note that the quadratic matrix inequality constraint in (4) can be reformulated into a linear constraint by the Schur complement or by taking a square root on both sides of the inequality. In the reformulated form the problem can be solved with general purpose semidefinite programming solvers like `SDPT3` or `CVXOPT`.

### 3.2 Kernel based model representation

The kernel based estimation problem given in Lemma 1 is not useful on its own, but only when a model representation in terms of the dual variable $\boldsymbol{A}$ can be established. The derivation of this representation requires establishing a link between the primal model parameters $\boldsymbol{W}$, $\boldsymbol{b}$ and the Lagrange multipliers $\boldsymbol{A}$. To obtain such a connection the set of all possible values for $\boldsymbol{W}$ corresponding to the optimal $\boldsymbol{C}$ in (3) is characterized in [3]. Based on this characterization one can find the following relations

$$\boldsymbol{W} = \boldsymbol{\Phi}\boldsymbol{A}^T\boldsymbol{M} \quad \text{and} \quad \boldsymbol{b} = \frac{1}{N}(\boldsymbol{Y} - \boldsymbol{M}\boldsymbol{A}\boldsymbol{\Omega})\boldsymbol{1}_N$$

with $\boldsymbol{M} = \eta^{-2}\boldsymbol{P}_\eta(\boldsymbol{Y}\boldsymbol{A}^T - \boldsymbol{A}\boldsymbol{A}^T)\boldsymbol{P}_\eta$ where $\boldsymbol{P}_\eta = \boldsymbol{V}_\eta\boldsymbol{V}_\eta^T$. The matrix $\boldsymbol{V}_\eta$ contains the eigenvectors corresponding to the largest eigenvalue of $\boldsymbol{A}\boldsymbol{\Omega}\boldsymbol{A}^T$, where $\lambda_{\max} = \eta^2$.

The dual model representation and the corresponding predictive model for a new point $\boldsymbol{z}$ are obtained by the substitution of these relations into the parametric model (1),

$$\widehat{\boldsymbol{y}} = \boldsymbol{f}(\boldsymbol{z}) = \sum_{t=1}^{N} \widetilde{\boldsymbol{\alpha}}_t K(\boldsymbol{x}_t, \boldsymbol{z}) + \boldsymbol{b}.$$

The variables $\widetilde{\boldsymbol{\alpha}}_t$ form the matrix $\widetilde{\boldsymbol{A}} = [\widetilde{\boldsymbol{\alpha}}_1, \ldots, \widetilde{\boldsymbol{\alpha}}_N]$, which is computed as $\widetilde{\boldsymbol{A}} = \boldsymbol{M}\boldsymbol{A}$.

## 4 Numerical example

Figure 1 compares the validation performance for different estimation problems on a toy data set. The data is generated using a model of the form $\boldsymbol{y}_t = \boldsymbol{W}_0^T\boldsymbol{\varphi}(\boldsymbol{x}_t) + \boldsymbol{e}_t$ with $M = 20$ outputs. The parameter matrix $\boldsymbol{W}_0$ is constructed as $\boldsymbol{W}_{0,B}\boldsymbol{W}_{0,M}$. $\boldsymbol{W}_{0,B}$ is a $50 \times 3$ matrix, while $\boldsymbol{W}_{0,M}$ is $3 \times 20$. The elements of both matrices are drawn from a standard normal distribution.

The figure compares the estimation technique described in this abstract, denoted by MIMO, with models with a simple quadratic regularization term $\boldsymbol{w}_i^T\boldsymbol{w}_i$ trained on each output denoted by RR and two ordinary least squares (OLS) estimates without any regularization.

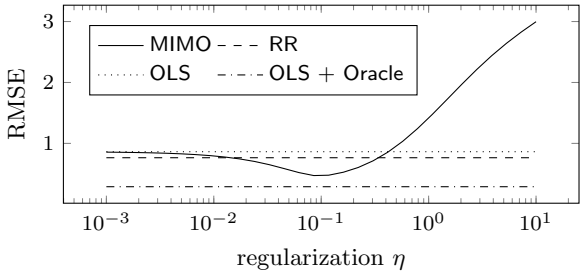



**Fig. 1:** Validation performance of different multivariate model structures for toy dataset. The predictive performance is shown as a function of the regularization parameter. The root mean squared error in this figure is computed with respect to time $t$ and output $i$.

The model with oracle was given the matrix $\boldsymbol{W}_{0,M}$ such that it only had to estimate the parameters of $\boldsymbol{W}_{0,B}$, which accounts to a factor of 60 less in the number of parameters. One can observe that the proposed scheme clearly outperforms OLS as well as RR. The knowledge of the true dependencies still yields a better performance though.

## 5 Conclusions

This contribution proposed a novel formulation to estimate kernel based models with multiple outputs in an identification setting and illustrated the effectiveness on a simple toy example. Furthermore the kernel based model representation for a model with nuclear norm regularization has been stated in a primal-dual setting.

**Acknowledgments**

This work was supported by CoE EF/05/006 OPTEC, GOA MANET, IUAP DYSCO, FWO G.0377.12, ERC AdG A-DATADRIVE-B. J.A.K. Suykens is a professor and B. De Moor is a full professor at KU Leuven, Belgium.

## References

[1] C. Alzate, M. Espinoza, B. De Moor, and J.A.K. Suykens. Identifying customer profiles in power load time series using spectral clustering. In *Proc. of the 19th International Conference on Artificial Neural Networks*, pages 315–324, 2009.

[2] A. Argyriou, C.A. Micchelli, and M.A. Pontil. When Is There a Representer Theorem? Vector Versus Matrix Regularizers. *Journal of Machine Learning Research*, 10:2507–2529, 2009.

[3] T. Falck. *Nonlinear system identification using structured kernel based modeling.* PhD thesis, Katholieke Universiteit Leuven, Belgium, April 2013.

[4] M. Fazel. *Matrix Rank Minimization with Applications.* PhD thesis, Stanford, 2002.

[5] J.A.K. Suykens, C. Alzate, and K. Pelckmans. Primal and dual model representations in kernel-based learning. *Statistics Surveys*, 4:148–183, August 2010.

[6] J.A.K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, and J. Vandewalle. *Least Squares Support Vector Machines.* World Scientific, 2002.

# First-order methods for low-rank matrix factorization applied to informed source separation

Augustin Lefèvre[1]
augustin.lefevre@uclouvain.be
[1] Université catholique de Louvain - ICTEAM Institute
Avenue Georges Lemaître 4, B-1348 Louvain-la-Neuve - Belgium

François Glineur[1,2]
francois.glineur@uclouvain.be
[2] Université catholique de Louvain - CORE
Voie du Roman Pays 34, B-1348 Louvain-la-Neuve - Belgium

**Abstract**: We study a convex formulation of low-rank matrix factorization, in a special case where additional information on the factors is known. Our formulation is typically adapted to source separation scenarii, where additional information on the sources may be provided by an expert. Our formulation promotes low-rank with a nuclear-norm based penalty. As it is non-smooth, generic first-order algorithms suffer from slow convergence rates. We study and compare several algorithms that fully exploit the structure of our problem while keeping memory requirements linear in the size of the problem.

**Keywords**: source separation,inverse problems,machine learning

## 1 Low-rank matrix factorization and informed source separation

Given a matrix of observations $Y \in \mathbb{R}^{F\times N}$, we assume that $Y$ is a sum of $G$ low-rank contributions perturbed by some noise, i.e. $Y \approx \sum_{g=1}^{G} X_g$ where $X_g \in \mathbb{R}_+^{F\times N}$ is the contribution of source $g$. The informed source separation problem consists in identifying contributions $X_g$ with the additional knowledge that some entries in some of the contributions are equal to zero.

Matrix factorization is an essential building block in source separation methods. Indeed, as we seek to represent $Y$ as a sum of low-rank matrices, an equivalent problem is to express $Y$ approximately as a low-rank product of factors $D \in \mathbb{R}^{F\times K}$ and corresponding activation coefficients $A \in \mathbb{R}^{K\times N}$ (where the inner size $K$ is the sum of the ranks of the contributions). More specifically, one seeks to minimize a suitable norm of the difference $\|Y - DA\|$. Unfortunately, the problem of identifying the best factors $D$ and $A$ is nonconvex and multimodal, so that in practice only local solutions can be expected to be found in reasonable time (i.e. estimates of $D$ and $A$ in the neighbourhood of which no improvement can be made).

## 2 Nuclear norm-based convex reformulation

In this work, we substitute the above nonconvex problem with a convex reformulation. A well-known technique to obtain low-rank estimates for the sources $X_g$ is to penalize $\|X_g\|_\star$, the nuclear norm of $X_g$, i.e. the sum of its singular values $\sigma_1 \geq \sigma_2 \geq \ldots \sigma_F$. This leads us to consider the following problem, introduced earlier in [1] :

$$\min_X \quad \tfrac{1}{2}\|Y - \textstyle\sum_{g=1}^{G} X_g\|_F^2 + \lambda \sum_{g=1}^{G} \|X_g\|_\star \quad (1)$$

$$\text{subject to} \qquad X_g \geq 0 \text{ and } M_g \cdot X_g = 0, \quad (2)$$

where $A\cdot B$ denotes the coefficientwise product of $A$ and $B$, and the matrices $M_g$ enforce the constraint that the entry of $X_g$ at coordinates $(f, n)$, $X_{g,fn} = 0$ if $M_{g,fn} \neq 0$. In what follows, we define a scalar product $\langle U, V\rangle = \sum_g \mathrm{Tr} U_g^\top V_g$ for the search space. Problem 2 is convex, because we have replaced the strict constraint that each source term $X_g$ be low-rank by a penalty term $\psi(X) = \lambda \sum_g \|X_g\|_\star$ favouring low-rank solutions. Convexity is desirable because it means that global solutions may be reached from any initial point, without recourse to extensive sampling of the search space $\mathbb{R}^{F\times N\times G}$.

## 3 Algorithms for nuclear norm minimization

Because of the nuclear norm penalty term, the objective function in Problem 2 is non-smooth. In previous experiments with a projected subgradient algorithm [1], we observed that satisfactory source separation could be obtained fast enough, provided the step size is carefully selected. We compare in this article algorithms

that enjoy faster convergence rates than subgradient, by exploiting the specific structure of the nuclear norm : indeed, the general non-smooth minimization method relies on an arbitrary choice of subgradient at each iteration, even though it might not be a descent direction. In our situation, the subdifferential of the objective function may be described completely, so we can make our choice more wisely, as we will show in Section 3.1. On the other hand, we also consider applying an optimal gradient method to a smooth approximation of the nuclear norm [3], which we will detail in Section 3.2.

### 3.1 Subgradient methods

Let $\partial f(X)$ be the subdifferential of $f$ at $X$. Since the objective function $f$ is convex, it also admits directional derivatives $f'(X;D)$ in every direction, and we have $f'(X;D) = \max\{\langle U, D\rangle, U \in \partial f(V)\}$. General subgradient methods for minimizing $f$ consist in successively picking a subgradient $G \in \partial f(V)$ at a given point $X$, moving $X$ along the direction $-G$ with an approprate choice of the step size, and projecting $X$ on the set of constraints, if any. However, while the (projected) gradient is always a descent direction when $f$ is differentiable, it is no longer the case for arbitrary choices of subgradients. Fortunately, the steepest descent direction is related to the minimum norm subgradient :

$$\arg\min\{f'(X;D)\} = -\arg\min\{\|Z\|, Z \in \partial f(X)\}$$

In the unconstrained case, the steepest descent direction is always a descent direction (provided, of course, $X$ is not a global minimum). This means that the step size can be chosen so as to ensure :$f(X + \alpha G) \leq f(X)$. In our case, additional care must be taken to select a *feasible* descent direction. As we show, computing the minimum norm subgradient implies roughly twice the computational cost of an arbitrary subgradient, so it is worth comparing its merit experimentally.

### 3.2 Smoothing based gradient methods

Nesterov [3] showed that for a particular class of non-smooth functions (in which our Problem fits), it is possible to obtain faster convergence rate by applying an accelerated gradient method to a smooth approximation $f_\mu$ of the objective function. In our case, we replace the nuclear norm term by :

$$\|X\|_{\star,\mu} = \sum_{f=1}^{F} h_\mu(\sigma_f) \text{ where } h_\mu(x) = \begin{cases} \frac{x^2}{2\mu} & \text{if } x \leq \mu \\ x - \frac{\mu}{2} & \text{if } x > \mu \end{cases}$$

When $\mu = 0$, $\|X\|_{\star,\mu} = \|X\|_\star$ and for general $\mu > 0$, it is a continuously differentiable function with Lipschitz continuous derivatives with the Lipschitz constant $\frac{1}{\mu}$. We show that the Lipschitz constant of $f_\mu$ is $L_\mu = G + \frac{\lambda}{\mu}$. Moreover, $f_\mu(X) \leq f(X) \leq f_\mu(X) + \frac{\mu}{2}$, so parameter $\mu$ trades off the magnitude of the Lipschitz constant $L_\mu$ (and hence the rate of convergence of the fast gradient algorithm) versus the quality of the approximation error. Following [3] we implement an accelerated gradient method with fixed step size. Values and first order derivatives of $\|X\|_{\star,\mu}$ are obtained by computing the SVD of $X$, so we can implement a fast gradient method with the same iteration cost as projected subgradient descent, but whose theoretical convergence rate is much faster.

### 3.3 Contributions and related work

Algorithms for the minimization of the nuclear norm subject to linear equality constraints have been proposed in [2, 4]. For small problems, the author show that nuclear norm minimization problems may be reformulated as semidefinite programs (SDP), for which interior-point algorithms with superlinear convergence are available. For the purpose of source separation, interior-point methods are too expensive as they require storage and inversion of matrices of size $(F \times N)^2$, where $F \simeq 500$, $N \simeq 10^3$. Another interesting category of algorithms are those based on augmented Lagrangians and explicit factorizations of the source terms $X_G D_g A_g$ [4].

## 4 Experimental results and comments

We compare smoothing based gradient methods and subgradient (with or without minimum norm subgradient) in an informed source separation experiment involving four musical pieces of 14 seconds each. For appropriate values of $\mu > 0$, smoothing based gradient methods achieve both faster decrease of the objective function, and better quality solutions, while the minimum norm sub-gradient improves much over an arbitrary one.

**Acknowledgments**

This paper presents research results of the Belgian Network DYSCO (Dynamical Systems, Control, and Optimization), funded by the Interuniversity Attraction Poles Programme, initiated by the Belgian State, Science Policy Office. The scientific responsibility rests with its authors.

## References

[1] A. Lefèvre, F. Glineur, and P.-A. Absil. A nuclear norm-based convex formulation for informed source separation. Technical Report 1212.3119, arXiv, 2012.

[2] Z. Liu and L. Vandenberghe. Interior-point method for nuclear norm approximation with application to system identification. *SIAM Journal on Matrix Analysis and Applications*, 2009.

[3] Yu. Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming*, 2005.

[4] B. Recht, M. Fazel, and P.A. Parrilo. Guaranteed Minimum-Rank Solutions of Linear Matrix Equations via Nuclear Norm Minimization. *SIAM Review.*, 2010.

# Structured low-rank approximation as optimization on a Grassmann manifold

Konstantin Usevich, Ivan Markovsky
Dept. ELEC, Vrije Universiteit Brussel
{kusevich, imarkovs}@vub.ac.be

**Abstract**: Many data modeling problems can be posed and solved as a structured low-rank approximation problem. Using the variable projection approach, the problem is reformulated as optimization on a Grassmann manifold. We compare local optimization methods based on different parametrizations of the manifold, including recently proposed penalty method and method of switching permutations. A numerical example of system identification is provided.

**Keywords**: structured low-rank approximation, variable projection, Grassmann manifold, local optimization, system identification

## 1 Introduction

A *linear structure* is a linear map $\mathbb{R}^{n_p} \to \mathbb{R}^{m\times n}$. For convenience, we assume that $m \leq n$. In this paper, we consider the problem of approximating a given structured matrix by a structured matrix of low rank.

**Problem 1** (Structured low-rank approximation)**.** *Given $p \in \mathbb{R}^{n_p}$, structure $\mathscr{S}$, seminorm $\|\cdot\|$ and $r < m$*

$$\underset{\widehat{p}\in\mathbb{R}^{n_p}}{\text{minimize}} \|\widehat{p} - p\| \quad \text{subject to} \quad \mathit{rank}\,\mathscr{S}(\widehat{p}) \leq r. \tag{1}$$

Many data modeling problems can be posed and solved as Problem 1, for a suitable structure, seminorm, and rank [4]. In this case, $p$ is an observed data vector, and the approximating vector $\widehat{p}$ should possess certain property, which is encoded in the low-rank constraint.

Usually, the weighted Euclidean seminorm is used

$$\|p\|_w^2 = \sum_{i=1}^{n_p} w_i p_i^2, \quad \text{where } w_i \in [0, +\infty]. \tag{2}$$

Infinite weights correspond to fixed (exactly known) values of the approximating vector [6] and zero weights correspond to the case of missing observations [5].

### 1.1 Variable projection solution

The rank constraint in (1) is equivalent to existence of a full row rank $R \in \mathbb{R}^{d\times m}$, where $d := m - r$, such that $R\mathscr{S}(\widehat{p}) = 0$. Hence Problem 1 can be reformulated as

$$\underset{R:\,\text{rank}\,R=d}{\text{minimize}} f(R), \quad \text{where} \tag{3}$$

$$f(R) := \min_{\widehat{p}\in\mathbb{R}^{n_p}} \|\widehat{p} - p\|_w^2 \quad \text{subject to} \quad R\mathscr{S}(\widehat{p}) = 0. \tag{4}$$

For the weighted semi-norm (2) the inner problem (4) has a closed-form solution [5]. For mosaic Hankel structure and positive weights there exist efficient $O(d^3m^2n)$ methods [6] for evaluating $f(R)$ and its derivatives.

The cost function is homogeneous in the following sense:

$$f(R) = f(UR) \quad \forall \text{ nonsingular } \quad U \in \mathbb{R}^{d\times d}. \tag{5}$$

Therefore, $f$ depends only on the row space of $R$. Hence, $f$ is defined on a *Grassmann manifold* $\text{Gr}_{\mathbb{R}}(d, m)$ [3] (the manifold of all $d$-dimensional subspaces of $\mathbb{R}^m$).

## 2 Optimization on $\text{Gr}_{\mathbb{R}}(d, m)$

Consider a homogeneous (in the sense of (5)) function $f : \mathcal{R}_{\mathtt{f}} \to \mathbb{R}_+$, where

$$\mathcal{R}_{\mathtt{f}} := \{R \in \mathbb{R}^{d\times m} : \text{rank}\, R = d\}, \tag{6}$$

and $\mathbb{R}_+ := [0; +\infty)$. We require the function $f$ to be smooth (from $C^1(\mathcal{R}_{\mathtt{f}})$). The optimization problem considered is (3), which is minimization on $\text{Gr}_{\mathbb{R}}(d, m)$.

### 2.1 Orthonormal bases

For any subspace there exists an orthonormal basis. Hence, (3) is equivalent to

$$\underset{R\in\mathbb{R}^{d\times m}}{\text{minimize}} f(R) \quad \text{subject to} \quad RR^\top = I_d. \tag{7}$$

Problem (7) can be solved by constrained optimization methods or by methods using Riemannian geometry [3].

### 2.2 Exact penalty method of [5] (reg)

In [5] it was shown that the hard constraint in (7) can be equivalently replaced by a soft constraint.

**Theorem 1.** *Let $\widetilde{f} : \mathbb{R}^{d\times m} \to \mathbb{R}_+$ be a homogeneous extension (not necessarily smooth) of* (4)*, i.e. $\widetilde{f}(R) = f(R)$ for all $R \in \mathcal{R}_{\mathtt{f}}$, and $\widetilde{f}$ satisfies* (5)*. Let $\gamma > f(R_0)$ for an $R_0 \in \mathcal{R}_{\mathtt{f}}$. Then the solutions of*

$$\underset{R\in\mathbb{R}^{d\times m}}{\text{minimize}} f(R) + \gamma\|RR^\top - I_d\|_{\text{F}}^2, \tag{8}$$

*coincide with the solutions of* (3)*.*

Therefore, the penalty method in (8) is exact.

| Parameters | | | | `perm`$_0$ | | | `perm` | | | `genrtr` | | | `fmincon` | | | `reg` | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $T$ | $q$ | $\mathtt{m}$ | $\ell$ | % | it | $t$ | % | it | $t$ | % | it | $t$ | % | it | $t$ | % | it | $t$ |
| 57 | 7 | 5 | 1 | 0.04 | 200 | 0.07 | 0.04 | 21 | 0.01 | 0.05 | 200 | 1.69 | 0.04 | 66 | 0.73 | 0.04 | 22 | 0.26 |
| 90 | 8 | 5 | 1 | 0.06 | 200 | 0.27 | 0.06 | 200 | 0.26 | 0.07 | 200 | 6.62 | 0.07 | 87 | 1.4 | 0.08 | 12 | 0.29 |
| 801 | 2 | 1 | 2 | 1.46 | 19 | 0.01 | 1.46 | 19 | 0.01 | 1.46 | 15 | 0.2 | 1.45 | 49 | 0.25 | 1.56 | 200 | 0.78 |
| 1,000 | 2 | 1 | 5 | 3.45 | 80 | 0.1 | 3.45 | 77 | 0.1 | 4.25 | 200 | 6.43 | 4.22 | 67 | 0.49 | 4.35 | 200 | 1.67 |
| 1,024 | 2 | 1 | 5 | 9.75 | 44 | 0.06 | 9.74 | 200 | 0.25 | 17.46 | 200 | 0.69 | 13.39 | 56 | 0.47 | 13.31 | 25 | 0.27 |

**Tab. 1:** Comparison of the methods

### 2.3 Switching between permutations (`perm`)

Any subspace from $\mathrm{Gr}_{\mathbb{R}}(d,m)$ can be represented by a matrix of the form $[\,X\ -I_d\,]\Pi$, where $\Pi$ is a permutation matrix and $X \in \mathbb{R}^{d\times(m-d)}$. As shown in [1], $\Pi$ can be chosen such that $|X_{i,j}| \leq 1$ for all $i$, $j$. Hence, problem (3) is equivalent to

$$\underset{\substack{\Pi\in\{0,1\}^{m\times m},\\ \Pi\Pi^\top = I_m}}{\text{minimize}} \ \min_{X\in[-1;1]^{d\times(m-d)}} f\left(\begin{bmatrix} X & -I_d \end{bmatrix}\Pi\right). \qquad (9)$$

For local optimization of (9), the combinatorial problem of choosing $\Pi$ can be avoided by switching permutations in the course of optimization [7].

### 2.4 Using Riemannian geometry [3] (`genrtr`)

Let $\mathcal{M}$ be a Riemannian manifold, with tangent bundle $T_x\mathcal{M}$. The methods of [3] require a *retraction* $R_x : T_x\mathcal{M} \to \mathcal{M}$, and operate as follows. From $x_k$, a direction $\xi_k \in T_{x_k}$ is selected, based on derivatives of $f$ at $x_k$. The new iterate is $x_{k+1} = R_{x_k}(\xi_k)$, and the process is repeated until a convergence criterion is met. A trust-region method is implemented in [8].

## 3 Results

We compare the methods on an example of errors-in-variable system identification. The data $w \in (\mathbb{R}^q)^T$ is a $q$-variate time series, and the structure is block-Hankel:

$$\mathscr{H}_{\ell+1}(w) := \begin{bmatrix} w(1) & w(2) & \cdots & w(T-\ell) \\ w(2) & w(3) & & \vdots \\ \vdots & & & \vdots \\ w(\ell+1) & \cdots & \cdots & w(T) \end{bmatrix},$$

where the number of rows is $m = q(\ell+1)$. Identification in the class $\mathscr{L}^q_{\mathtt{m},\ell}$ of linear time-invariant systems with at most $\mathtt{m}$ inputs and lag at most $\ell$ is equivalent to Problem 1 for the structure $\mathscr{H}_{\ell+1}(w)$ and $r = m - (q - \mathtt{m})$ [4] (rank reduction by the number of outputs).

We compare the methods on examples from DAISY database [2]. In Table 1, we provide for each method the number of iterations (it), time ($t$), and the fit (%), equal to $100\% \cdot \|\widehat{p}_* - p\|_2^2 / \|p\|_2^2$, where $\widehat{p}_*$ is the computed approximation. `fmincon` stands for the minimization in orthonormal bases with the Optimization Toolbox of MATLAB. `perm`$_0$ denotes `perm` with $\Pi$ fixed to $I_m$. The methods are given the same initial approximation. In `perm` and `perm`$_0$, the Levenberg-Marquardt method is used for local optimization over $X$. Our preliminary results suggest that methods `perm` and `reg` are competitive with the method `genrtr` (see [7] for more details).

**Acknowledgments**

The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) / ERC Grant agreement no. 258581 "Structured low-rank approximation: Theory, algorithms, and applications".

## References

[1] D. E. Knuth. 'Semi-optimal bases for linear dependencies'. *Lin. and Multilin. Algebra* **17**, 1–4, 1985.

[2] B. De Moor, P. De Gersem, B. De Schutter and W. Favoreel. 'Daisy : A database for identification of systems'. *Journal A (Benelux publication of the Belg. Fed. of Aut. Control)* **38**, 4–5, 1997.

[3] P.-A. Absil, R. Mahony and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press. Princeton, NJ, 2008.

[4] I. Markovsky. *Low Rank Approximation: Algorithms, Implementation, Applications*. Communications and Control Engineering. Springer, 2012.

[5] I. Markovsky and K. Usevich. 'Structured low-rank approximation with missing values'. *SIAM J. Matrix Anal. Appl.*, in press, 2013.

[6] K. Usevich and I. Markovsky. 'Variable projection methods for affinely structured low-rank approximation in weighted 2-norms', *J. Comput. Appl. Math.*, 2013. doi:10.1016/j.cam.2013.04.034

[7] K. Usevich and I. Markovsky. 'Structured low-rank approximation as optimization on a Grassmann manifold', preprint, 2013. Available from: `http://homepages.vub.ac.be/~kusevich/preprints.html`.

[8] *GenRTR (Generic Riemannian Trust-Region package)*. Available from: `http://www.math.fsu.edu/~cbaker/genrtr/`.

# Scalable Structured Low Rank Matrix Optimization Problems

Marco Signoretto
ESAT-SCD/SISTA Katholieke Universiteit Leuven,
Kasteelpark Arenberg 10, B-3001 Leuven (BELGIUM).
marco.signoretto@esat.kuleuven.be

Volkan Cevher
Laboratory for Information and Inference Systems (LIONS),
EPFL STI IEL LIONS ELD 243 Station 11,
Lausanne (SWITZERLAND). volkan.cevher@epfl.ch

Johan A. K. Suykens
ESAT-SCD/SISTA Katholieke Universiteit Leuven,
Kasteelpark Arenberg 10, B-3001 Leuven (BELGIUM).
johan.suykens@esat.kuleuven.be

**Abstract**: We consider a class of structured low rank matrix optimization problems. We represent the desired structure by a linear map, termed mutation, that can encode matrices having entries partitioned into known disjoined groups. Our interest arises in particular from concatenated block-Hankel matrices that appear in formulations for input-output linear system identification problems with noisy and/or partially unobserved data. We present an algorithm and test it against an existing alternative.

## 1 Introduction

Nuclear norm optimization methods for low-rank matrix approximation have been discussed in several recent papers. In many different settings, in fact, general notions of model complexity can be conveniently expressed by the rank of an appropriate matrix; in turn, well known properties of the nuclear norm motivate its used for convex relaxations of rank-based problems. In this contribution, we focus on problems where we need to find a matrix that, in addition to being low-rank, is required to have entries partitioned into known disjoint groups. This setting includes various type of structured matrices such as Hankel, Toeplitz and circulant matrices. Generally speaking, it allows to deal with matrices that have reduced degrees of freedom. The desired structure is encoded by a linear map, termed *mutation*, that we characterize and use in our algorithm. We then discuss an application to linear system identification problems with noisy and partially unobserved input-output data. A recently proposed algorithm that applies to this problem involves at each iteration the singular value decomposition (SVD) of a structured matrix $Y \in \mathbb{R}^{M\times N}$, with $M \leq N$. The cost of this step usually grows as $O(M^2N)$ which hinders the application to large scale problems.

## 2 Problem Statement

We present an SVD-free solution strategy that delivers low rank solutions with comparable quality (in terms of objective value, feasibility and model fit) at a substantially lower computational price.

### 2.1 Main Problem Formulation

Recall that, for a generic matrix $A \in \mathbb{R}^{M\times N}$ with rank $R$ and singular values $\sigma_1(A) \geq \sigma_2(A) \geq \cdots \geq \sigma_R(A) > 0$, the *nuclear norm* (a.k.a. trace norm or Schatten-1 norm) is defined as $\|A\|_* = \sum_{r\in\mathbb{N}_R} \sigma_r(A)$ where $\mathbb{N}_R$ is used as a shorthand for the set $\{1, 2, \ldots, R\}$. We focus on the situation where we need to find a matrix that, in addition to being low-rank, is required to have entries partitioned into known disjoint groups. For a significant class of problems this task can be accomplished solving the convex optimization problem:

$$\min_{x\in\mathbb{R}^L} \frac{1}{2}(x-a)^\top H(x-a) + \|\mathcal{B}(x)\|_* \qquad (1)$$

for a given vector $a \in \mathbb{R}^L$, a positive definite matrix $H \in \mathbb{R}^{L\times L}$ and a linear map $\mathcal{B}$, the features of which are given below. The corresponding quadratic term usually plays the role of a data fitting measure and can subsume a trade-off parameter $\lambda$. A conventional choice is, in particular, $H = \lambda I_L$, where $I_L$ is the $L \times L$ identity matrix.

### 2.2 Encoding the Structure by Mutations

Formulation (1) regards a structured matrix as the output of a linear map $\mathcal{B} : \mathbb{R}^L \to \mathbb{R}^{M\times N}$, termed *mutation*, that maps entries of a vector $x$ into disjoint groups $\mathscr{P}_l$, $l = 1, 2, \ldots, L$ forming a partition of the set of entries of the matrix $\mathcal{B}(x)$. More formally, if $\iota : (m, n) \mapsto \{l \ : \ (m, n) \in \mathscr{P}_l\}$ is a well defined membership function, then:

$$\mathcal{B} : x \mapsto \left(x_{\iota(m,n)} \ : \ (m, n) \in \mathbb{N}_M \times \mathbb{N}_N\right) . \qquad (2)$$

Mutations give rise to structured matrices with Hankel, Toeplitz and circulant matrices as special cases. It is

not difficult to show that the adjoint operator is:

$$\mathcal{B}^* : \quad X \quad \mapsto \quad \left( \sum_{(m,n)\in\mathcal{P}_l} x_{mn} \ : \ l \in \mathbb{N}_L \right) . \qquad (3)$$

From (2) and (3) one can then show that $\mathcal{B}^*\mathcal{B}$ is represented by a diagonal matrix with entries equal to the cardinality of the sets $\mathscr{P}_l$, $l = 1, 2, \ldots, L$. Starting from a known partition one can then give a "quick and dirty" implementation of these operators based on linear indexing [5]. More efficient implementations can be given for mutations encoding matrices with more specialized structure.

## 3 SVD-free Algorithm via Augmented Lagrangian Approach

In order to avoid computing SVDs we rely on a known variational characterization of the nuclear norm:

$$\|Y\|_* = \arg \min_{U,V \ : \ Y=UV^\top} \frac{1}{2} \left( \|U\|_F^2 + \|V\|_F^2 \right) \qquad (4)$$

where we considered full (i.e., not thin) matrices $U$ and $V$. This suggests the following restatement of problem (1):

$$\begin{array}{ll} \min\limits_{x,U,V} & \frac{1}{2}(x-a)^\top H(x-a) + \frac{1}{2} \left( \|U\|_F^2 + \|V\|_F^2 \right) \\ \text{subject to} & \mathcal{B}(x) = UV^\top \ . \end{array} \qquad (5)$$

Note that this problem is non-convex due to the product between $U$ and $V$. Nonetheless, it is possible to show that if $(x_*, U_*, V_*)$ is a solution to (5), $x_*$ is a solution to (1) and $\mathcal{B}(x_*) = U_* V_*^\top$. A similar result is found in [4, Section 5.3], which discusses an approach based on (4) to solve an unstructured low rank matrix problem. Problem (5) has a differentiable objective function. It can be tackled via a gradient-based solution strategy after embedding the constraint $\mathcal{B}(x) = UV^\top$ by an augmented Lagrangian approach. Details of our approach are given in [5]. Note that, in contrast, the objective of problem (1) is non-smooth. Correspondingly, computing a solution to problem (1) requires a sub-gradient approach or an operator splitting technique [1]. This, ultimately, leads to *singular values soft-thresholding* [2] which requires computing at each iteration the SVD of an $M \times N$ matrix.

## 4 System Identification with Missing Data by Nuclear Norm Optimization

Recently, [3] proposed the following formulation for linear system identification from input-output data $\tilde{u}$, $\tilde{y}$:

$$\min_{u,y} \frac{\lambda_1}{2} \sum_{i\in\mathscr{M}_u} (u_i - \tilde{u}_i)^2 + \frac{\lambda_2}{2} \sum_{i\in\mathscr{M}_y} (y_i - \tilde{y}_i)^2 + \|F(u,y)\|_* \qquad (6)$$

where $u$, $y$ are latent inputs and outputs, $\mathscr{M}_u$ (resp. $\mathscr{M}_y$) is a set of indices of observed inputs (resp. outputs) and $F(u,y)$ is a matrix obtained stacking block-Hankel matrices $\mathcal{H}_Q(u)$ and $\mathcal{H}_Q(y)$. This method is motivated by the fact that, if the generating dynamical system is linear, then under mild conditions one has $\text{rank}(F(u,y)) = S + \text{rank}(\mathcal{H}_Q(u))$ where $S$ is the order of the system. It is not difficult to show that this problem is equivalent to (1) for suitably defined mutation $\mathcal{B}$ and diagonal p.s.d. matrix $H$ [5].

**Tab. 1:** Results for $V\% = 20$, $P = 2$, $S = 3$.

| | obj val | feas. ($10^{-3}$) | model fit | CPU time |
|---|---|---|---|---|
| | $M = 5,\ T = 1500$ | | | |
| SVD-free | 1016.79 | 0.30 | 79 | 0.67 |
| SVD-based | 1017.36 | 0.87 | 79 | 5.61 |
| | $M = 15,\ T = 1500$ | | | |
| SVD-free | 1166.05 | 0.51 | 75 | 1.96 |
| SVD-based | 1165.71 | 0.30 | 75 | 8.65 |
| | $M = 20,\ T = 4000$ | | | |
| SVD-free | 2079.09 | 0.22 | 76 | 6.78 |
| SVD-based | 2081.47 | 0.86 | 76 | 47.01 |
| | $M = 40,\ T = 4000$ | | | |
| SVD-free | 2501.61 | 0.39 | 70 | 22.32 |
| SVD-based | 2501.76 | 0.37 | 70 | 120.16 |
| | $M = 50,\ T = 10000$ | | | |
| SVD-free | 4803.46 | 0.30 | 67 | 111.28 |
| SVD-based | 4803.18 | 0.92 | 67 | 825.77 |

## 5 Experiments

We tested the SVD-free approach presented in [5] against the SVD-based algorithm of [3] on randomly generated linear dynamical systems; details are given in [5]. Table 1 reports a representative set of experiments in which we kept fixed the percentage of missing data ($V\%$), the input dimension ($P$), the system order ($S$) and vary the output dimension ($M$) and the number of samples ($T$). The experiments show that the proposed algorithm delivers low rank solutions with comparable quality (in terms of objective value, feasibility and model fit) at a substantially lower computational price than the existing SVD-based algorithm.

## References


[1] H.H. Bauschke and P.L. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer Verlag, 2011.

[2] J.F. Cai, E.J. Candès, and Z. Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010.

[3] Z. Liu, A. Hansson, and L. Vandenberghe. Nuclear norm system identification with missing inputs and outputs. *Submitted to System and Control Letters*, 2012.

[4] B. Recht, M. Fazel, and P.A. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Rev.*, 52:471–501, 2010.

[5] M. Signoretto, V. Cevher, and Suykens J. A. K. An SVD-free approach to a class of structured low rank matrix optimization problems with application to system identification. Technical report, 13-44, ESAT-SISTA, K.U.Leuven (Leuven, Belgium), Submitted, 2013.

# Learning with Marginalized Corrupted Features

Laurens van der Maaten
Delft Univ. of Techn.
lvdmaaten@gmail.com

Minmin Chen
Wash. Univ. in St. Louis
mc15@cec.wustl.edu

Stephen Tyree
Wash. Univ. in St. Louis
swtyree@wustl.edu

Kilian Weinberger
Wash. Univ. in St. Louis
kilian@wustl.edu

**Abstract**: We propose a new framework for regularization, called marginalized corrupted features, that reduces overfitting by increasing the robustness of the model to data corruptions.

**Keywords**: Regularization; Supervised learning.

## 1 Introduction

Dealing with overfitting is one of the key problems one encounters when training machine-learning models. Three approaches are commonly used to combat overfitting: (1) *early-stopping* techniques stop the learning as soon as the performance on a held-out validation set deteriorates; (2) *regularizers* encourage the learning to find "simple" models by penalizing "complex" models, *e.g.*, models with large parameter values; and (3) *Bayesian* techniques define a prior distribution over models that favor simple models, and perform predictions by averaging over the model posterior.

We propose an alternative to counter overfitting, called *marginalized corrupted features* (MCF; [1]). Instead of requiring the user to define priors over *model parameters*, which can be very counter intuitive, we focus on corruptions of the *data*. MCF is based on the observation that overfitting would completely disappear if we were to train on infinite data drawn from the data distribution $\mathcal{P}$. Unfortunately, a learning scenario in which we only obtain a finite training set is more realistic. In many learning scenarios, we may however have some additional knowledge about the data distribution: we might know that certain *corruptions* of data instances do not affect their label. As an example, deleting a few words in a text document rarely changes its topic. With this prior knowledge, we can corrupt existing data to generate new artificial instances that resemble those sampled from the actual data distribution. MCF *corrupts* the existing *finite* training examples with a fixed corrupting distribution to construct an *infinite* training set in which the model is trained. For a wide range of learning models and corrupting distributions, we show that it is practical to train models on such an infinite, augmented training set.

## 2 Marginalized Corrupted Features

We start by defining a corrupting distribution that specifies how training observations $\mathbf{x}$ are transformed into corrupted versions $\tilde{\mathbf{x}}$. We assume that the corrupting distribution factorizes over dimensions and that each individual distribution $P_E$ is a member of the natural exponential family:

$$p(\tilde{\mathbf{x}}|\mathbf{x}) = \prod_{d=1}^{D} P_E(\tilde{x}_d|x_d;\eta_d),$$

where $\eta_d$ represents user-defined hyperparameters of the corrupting distribution on dimension $d$. Corrupting distributions of interest, $P_E$, include: (1) independent salt or "blankout" noise in which the $d$-th feature is randomly set to zero with probability $q_d$; (2) independent Gaussian noise on the $d$-th feature with variance $\sigma_d^2$; and (3) independent Poisson corruptions in which the $d$-th feature is used as the rate of the distribution.

Assume we are provided with a training data set $\mathcal{D} = \{(\mathbf{x}_n, y_n)\}_{n=1}^N$ and a loss function $L(\mathbf{x}, y; \Theta)$, with model parameters $\Theta$. A simple approach to approximately learn from the distribution $p(\tilde{\mathbf{x}}|\mathbf{x})\mathcal{P}(\mathbf{x})$, is to corrupt each training sample $M$ times, and to train on the resulting corrupted data in an empirical risk minimization framework, by minimizing:

$$\mathcal{L}(\tilde{\mathcal{D}};\Theta) = \sum_{n=1}^{N} \frac{1}{M} \sum_{m=1}^{M} L(\tilde{\mathbf{x}}_{nm}, y_n; \Theta),$$

with $\tilde{\mathbf{x}}_{nm} \sim p(\tilde{\mathbf{x}}_{nm}|\mathbf{x}_n)$. Although such an approach is effective, it lacks elegance and comes with high computational costs: the minimization of $\mathcal{L}(\tilde{\mathcal{D}};\Theta)$ scales linearly in the number of corrupted observations. MCF addresses these issues by considering the limiting case $M \to \infty$, in which we can rewrite $\frac{1}{M}\sum_{m=1}^{M} L(\tilde{\mathbf{x}}_m, y_m; \Theta)$ as its expectation to obtain:

$$\mathcal{L}(\mathcal{D};\Theta) = \sum_{n=1}^{N} \mathbb{E}[L(\tilde{\mathbf{x}}_n, y_n; \Theta)]_{p(\tilde{\mathbf{x}}_n|\mathbf{x}_n)}.$$

For linear predictors that employ a quadratic or exponential loss function, the required expectation can be computed analytically for all corrupting distributions in the natural exponential family; for linear predictors that employ logistic loss, we derive a practical upper bound on the expected loss. As a result, direct minimization of the resulting MCF loss functions is efficient.

**Quadratic loss.** Assuming a linear model parametrized by vector $\mathbf{w}$ and a target variable $y$ (for regression, $y$ is continuous; for binary classification, $y \in \{-1, +1\}$), the expected value of the quadratic loss under corrupting distribution $p(\tilde{\mathbf{x}}|\mathbf{x})$ is given by:

$$\begin{aligned}\mathcal{L}(\mathcal{D};\mathbf{w}) &= \sum_{n=1}^{N} \mathbb{E}\left[\left(\mathbf{w}^{\mathrm{T}}\tilde{\mathbf{x}}_n - y_n\right)^2\right]_{p(\tilde{\mathbf{x}}_n|\mathbf{x}_n)} \\ &= \mathbf{w}^{\mathrm{T}}\mathbf{H}\mathbf{w} - 2\left(\sum_{n=1}^{N} y_n \mathbb{E}\left[\tilde{\mathbf{x}}_n\right]\right)^{\mathrm{T}}\mathbf{w} + N,\end{aligned}$$

where the hat matrix $\mathbf{H} = \sum_{n=1}^{N} \mathbb{E}[\tilde{\mathbf{x}}_n]\mathbb{E}[\tilde{\mathbf{x}}_n]^{\mathrm{T}} + V[\tilde{\mathbf{x}}_n]$, $V[x]$ is the variance of $x$, and all expectations are under $p(\tilde{\mathbf{x}}_n|\mathbf{x}_n)$. Hence, to minimize the expected quadratic loss under the corruption, we only need to compute the mean and variance of the corrupting distribution.

**Exponential loss.** Assuming a label variable $y \in \{0, 1\}$, the expected value of the exponential loss under corruption model $p(\tilde{\mathbf{x}}|\mathbf{x})$ is given by:

$$\begin{aligned}\mathcal{L}(\mathcal{D};\mathbf{w}) &= \sum_{n=1}^{N} \mathbb{E}\left[\exp\left(-y_n \mathbf{w}^{\mathrm{T}}\tilde{\mathbf{x}}_n\right)\right]_{p(\tilde{\mathbf{x}}_n|\mathbf{x}_n)} \\ &= \sum_{n=1}^{N}\prod_{d=1}^{D} \mathbb{E}\left[\exp\left(-y_n w_d \tilde{x}_{nd}\right)\right]_{p(\tilde{x}_{nd}|x_{nd})}.\end{aligned}$$

The equation can be recognized as a product of moment-generating functions (MGFs), $\mathbb{E}[\exp(t_{nd}\tilde{x}_{nd})]$ with $t_{nd} = -y_n w_d$. MGFs can be computed for all corrupting distributions in the natural exponential family.

**Logistic loss.** The expected logistic loss cannot be computed in closed form. Instead, we derive an upper bound that can be minimized as a surrogate loss:

$$\begin{aligned}\mathcal{L}(\mathcal{D};\mathbf{w}) &= \sum_{n=1}^{N} \mathbb{E}\left[\log\left(1 + \exp\left(-y_n \mathbf{w}^{\mathrm{T}}\tilde{\mathbf{x}}_n\right)\right)\right]_{p(\tilde{\mathbf{x}}_n|\mathbf{x}_n)} \\ &\leq \sum_{n=1}^{N} \log\left(1 + \prod_{d=1}^{D} \mathbb{E}\left[\exp\left(-y_n w_d \tilde{x}_{nd}\right)\right]_{p(\tilde{x}_{nd}|x_{nd})}\right).\end{aligned}$$

Herein, we have made use of Jensen's inequality to upper-bound $\mathbb{E}[\log(z)]$. We again recognize a product of MGFs that can be computed in closed-form for corrupting distributions in the natural exponential family.

## 3 Experiments

**Document classification.** We perform experiments with blankout MCF and Poisson MCF on four Amazon data sets (using all three loss functions). All data sets have about 5,000 examples and 20,000 (bag-of-words) features. All our classifiers use L2-regularization in addition to MCF, and we cross-validated over the regularization parameter $\lambda$. The results of the experiments are shown in Figure 1. The results reveal the potential of MCF to improve the performance of linear predictors (note that "standard" classifiers correspond to blankout MCF with $q = 0$). The best performances are obtained using blankout MCF with $q \approx 0.7$ and $\lambda = 0$.

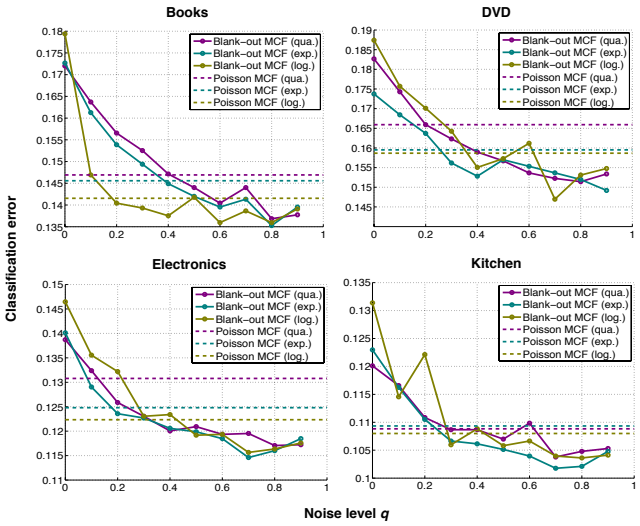



**Fig. 1:** Results of the document classification experiment.

**Nightmare at test time.** We experiment with blankout MCF on a *"nightmare at test time"* scenario in which a percentage of the features is randomly unobserved at test time on the MNIST handwritten digit data set. Figure 2 presents the classification error of standard and MCF classifiers as a function of the percentage of features deleted from the test data. The results show the strong performance of MCF in this scenario; in particular, MCF outperforms the state-of-the-art technique for this learning setting, FDROP.

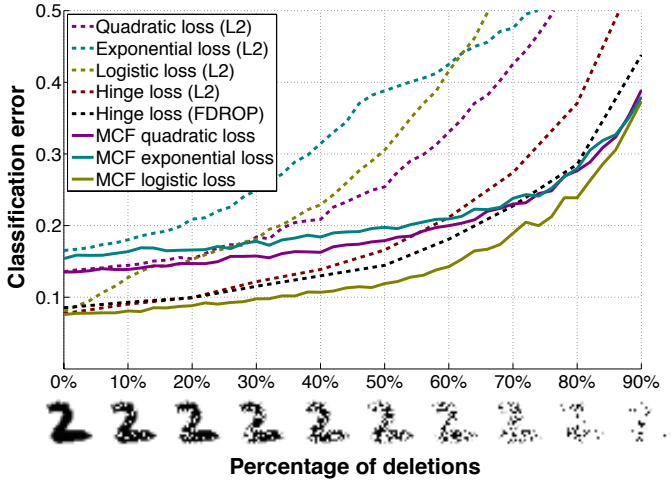



**Fig. 2:** Results of the *"nightmare at test time"* experiment.

## References

[1] L.J.P. van der Maaten, M. Chen, S.W. Tyree, and K.Q. Weinberger. Learning with Marginalized Features. *Int. Conf. on Machine Learning*, 2013.

# Robust regularized $M$-estimators of regression parameters and covariance matrix

Esa Ollila, Hyon-Jung Kim and Visa Koivunen
Department of Signal Processing and Acoustics, SMARAD CoE
Aalto University, Finland

**Abstract**: High dimension low sample size (HD-LSS) data are becoming increasingly present in a variety of fields, including chemometrics and medical imaging. Especially problems with $n < p$ (more variables than measurements) present a challenge to data analysts since the classical techniques can not be used. In this paper, we consider HD-LSS data in regression parameter and covariance matrix estimation problems. In particular, we consider and extend convex relaxation (or shrinkage regularization, diagonal loading) approach for $M$–estimation of regression coefficients and covariance (scatter) matrix. We demonstrate the utility of the methods in beamforming and tensor decomposition applications

**Keywords**: Covariance matrix estimation, Diagonal loading, High-dimensional data, $M$-estimator, Regularization, Ridge regression, Shrinkage

## 1 Introduction

High dimensional data sets are challenging for data analyst. In regression one often resorts to convex-relaxation methods such as ridge regression which seeks a balance with bias-variance trade-off. HD-LSS data provides a challenge to classical multivariate analysis as well. For example, principal component analysis (PCA) is a common pre-processing and standardization step which cannot be performed due to possibly rank deficient sample covariance matrix (SCM). Also impulsive measurement environments and outliers are commonly encountered in many practical applications. In this paper we tackle these two important issues and consider robust shrinkage approaches for regression parameter and covariance (scatter) matrix estimation problems in case of HD-LSS data.Recently robust shrinkage approaches for covariance matrix estimation were addressed in [1, 7] and for regression setting in [4].

## 2 Shrinkage $M$-estimates of regression

We consider the multiple linear regression model $y_i = \boldsymbol{\phi}_i^\top \mathbf{s} + \varepsilon_i$, $i = 1, \ldots, n$, which can be more conveniently expressed in matrix form as $\mathbf{y} = \boldsymbol{\Phi}\mathbf{s} + \boldsymbol{\varepsilon}$, where $\mathbf{y} = (y_1, \ldots, y_n)^\top$ is the observed data vector (*measurements*), $\boldsymbol{\Phi} = (\boldsymbol{\phi}_1 \cdots \boldsymbol{\phi}_n)^\top$ is the known $n \times p$ *measurement matrix* , $\mathbf{s} = (s_1, \ldots, s_p)^\top$ is the unobserved *signal vector* (or regression coefficients) and $\boldsymbol{\varepsilon} = (\varepsilon_1, \ldots, \varepsilon_n)^\top$ is the (unobserved) *noise* vector. The problem is then of estimating the unknown $\mathbf{s}$. All measurements and parameters are assumed to be real-valued. Suppose $\varepsilon_i$'s are i.i.d. from a continuous symmetric distribution with p.d.f. $f_\varepsilon(e) = \frac{1}{\sigma} f_0\big(\frac{e}{\sigma}\big)$, where $\sigma > 0$ denotes the *scale parameter* and $f_0(\cdot)$ the standard form of the p.d.f.

### 2.1 Ridge $M$-estimates with preliminary scale

We first assume that scale parameter $\sigma$ is known or replaced by its estimate. In either case, we replace hereafter $\sigma$ by $\hat{\sigma}$. First recall that ridge regression (RR) estimator [2] is defined as the (unique) minimizer of the penalized residual sum of squares objective function $J_{\mathrm{RR}}(\mathbf{s}) = \sum_{i=1}^n (y_i - \boldsymbol{\phi}_i^\top \mathbf{s})^2 + \lambda \|\mathbf{s}\|_2^2$, where $\lambda > 0$ is the *ridge (shrinkage or regularization) parameter*. The bigger the $\lambda$, the greater is the amount of shrinkage of coefficients toward zero. Let us denote *residuals* for a given (candidate) $\mathbf{s}$ by $e_i(\mathbf{s}) = y_i - \boldsymbol{\phi}_i^\top \mathbf{s}$ and write $\mathbf{e}(\mathbf{s}) = (e_1(\mathbf{s}), \ldots, e_n(\mathbf{s}))^\top$. We define the *ridge regression* $M$-estimator $\hat{\mathbf{s}}_\lambda$ as the minimizer of

$$J(\mathbf{s}) = \sigma^2 \sum_{i=1}^n \rho\bigg(\frac{e_i(\mathbf{s})}{\hat{\sigma}}\bigg) + \lambda \|\mathbf{s}\|_2^2 \qquad (1)$$

where $\rho$ is continuous and differentiable even function ($\rho(e) = \rho(-e)$) and increasing for $e \geq 0$. Note that the multiplying factor $\sigma^2$ is used so that the objective function coincide with $J_{\mathrm{RR}}(\mathbf{s})$ when $\rho(e) = e^2$.

Let us write $\psi(e) = \rho'(e)$ and $w(e) = \psi(e)/e$ with convention that $w(e) = 0$ for $e = 0$. To obtain robust RR $M$-estimates, we need $\rho$-functions that give small or zero weights for large residuals. In the conventional regression model ($n > p$), the maximum likelihood (ML) estimator of $\mathbf{s}$ is found by choosing $\rho(e) \propto -\log f_0(e) + c$ in (1) with $\lambda = 0$. In case of Cauchy error terms, $\rho_{\mathrm{C}}(e) = \frac{1}{2}\log(1 + e^2)$ and $\psi_{\mathrm{C}}(e) = \rho'(e) = e/(1 + e^2)$ whereas Huber's $\rho$ function is

$$\rho_{\mathrm{H}}(e) = \begin{cases} \frac{1}{2}e^2, & \text{for } |e| \leq k \\ k|e| - \frac{1}{2}k^2, & \text{for } |e| > k \end{cases}$$

and the corresponding $\psi$-function is $\psi_{\mathrm{H}}(e) =$

$\max[-k, \min(k, e)]$. Above $k$ is a user-defined *tuning constant* that affects robustness and efficiency of the method. With Huber's $\rho$-function, the objective function in (1) is convex but due to the non-convexity of Cauchy $\rho$, also the associated optimization problem is non-convex.

*Computation:* By setting the derivatives of (1) to zero shows that $\hat{\mathbf{s}}_\lambda$ solves the following estimating equation:

$$(\mathbf{\Phi}^\top \mathbf{W} \mathbf{\Phi} + 2\lambda \mathbf{I})\hat{\mathbf{s}}_\lambda = \mathbf{\Phi}^\top \mathbf{W} \mathbf{y} \tag{2}$$

where $\mathbf{W} = \mathrm{diag}(\{w_i\}_1^n)$ with $w_i = w\big(\mathbf{e}_i(\hat{\mathbf{s}}_\lambda)/\hat{\sigma}\big)$. This suggest the computation of the estimator by "iteratively (re)weighted RR (IWRR) algorithm. which iterates

$$\mathbf{s}_{t+1} = (\mathbf{\Phi}^\top \mathbf{W}_t \mathbf{\Phi} + 2\lambda \mathbf{I})^{-1} \mathbf{\Phi}^\top \mathbf{W}_t \mathbf{y} \tag{3}$$

until convergence. Note that an initial estimate $\mathbf{s}_0$ of $\mathbf{s}$ and an estimate $\hat{\sigma}$ of the scale of the residuals is needed. Following [4] it can be shown that the objective function (1) descends at each iteration. Thus for convex problems the IWRR algorithm can be used to find the global minimum.

### 2.2 Ridge $M$-estimates of regression and scale

Next we consider joint estimation of regression parameter $\mathbf{s}$ and scale $\sigma$. We define the joint ridge $M$-estimators of regression $\hat{\mathbf{s}}_\lambda$ and scale $\hat{\sigma}$ as the minimizers of

$$J(\mathbf{s}, \sigma) = \sigma \sum_{i=1}^{n} \rho\bigg(\frac{e_i(\mathbf{s})}{\sigma}\bigg) + \lambda \|\mathbf{s}\|_2^2 + \alpha(n\sigma). \tag{4}$$

where $\rho$ and $\lambda$ are as earlier and $\alpha \geq 0$ is a tuning parameter. Note that if $\lambda = 0$, then the equation reduces to approach proposed by Huber. By setting the derivatives of (4) w.r.t. $(\mathbf{s}, \sigma^2)$ to zero shows that $(\hat{\mathbf{s}}_\lambda, \hat{\sigma})$ solves the estimating equation (2) jointly with the estimating equation

$$\frac{1}{n} \sum_{i=1}^{n} \chi\Big(\frac{e_i(\hat{\mathbf{s}}_\lambda)}{\hat{\sigma}}\Big) = \alpha \tag{5}$$

where $\chi(e) = \psi(e)e - \rho(e)$ and $\psi = \rho'$ as earlier.

## 3 Shrinkage $M$-estimators of covariance

Here we consider the complex-valued case. Recall that a random vector (r.v.) $\mathbf{z} \in \mathbb{C}^p$ is said to have a $p$-variate *complex elliptically symmetric (CES) distribution* [5] if its p.d.f. is of the form $f(\mathbf{z}) = C_{p,g}|\mathbf{\Sigma}|^{-1} g\big(\mathbf{z}^{\mathrm{H}} \mathbf{\Sigma}^{-1} \mathbf{z}\big)$, for some positive definite Hermitian (PDH) $p \times p$ *scatter matrix* parameter $\mathbf{\Sigma}$ and function $g : \mathbb{R}_0^+ \to \mathbb{R}^+$, called the *density generator*; We shall write $\mathbf{z} \sim \mathrm{CE}_p(\mathbf{0}, \mathbf{\Sigma}, g)$. Above $C_{p,g}$ is a normalizing constant and $(\cdot)^{\mathrm{H}}$ denotes Hermitian transpose. Note that the covariance matrix of $\mathbf{z}$ (when exists) is equal to $\mathbb{E}[\mathbf{z}\mathbf{z}^{\mathrm{H}}] = c \cdot \mathbf{\Sigma}$. Consider $n$ i.i.d. samples from a CES distribution. Let us add a regularization term $\lambda \mathrm{Tr}(\mathbf{\Sigma}^{-1})$ to $-1\times$ log-likelihood function and minimize

$$L(\mathbf{\Sigma}) = \sum_{i=1}^{n} \rho(\mathbf{z}_i^{\mathrm{H}} \mathbf{\Sigma}^{-1} \mathbf{z}_i) + n \ln |\mathbf{\Sigma}| + \lambda \mathrm{Tr}(\mathbf{\Sigma}^{-1}). \tag{6}$$

where $\rho(t) = -\ln g(t)$ and $\lambda > 0$ is the fixed regularization parameter. Thus we impose a bound on $\mathrm{Tr}(\mathbf{\Sigma}^{-1}) = \sum_{i=1}^{p} \frac{1}{\gamma_i}$, where $\gamma_i$'s denote the eigenvalues of $\mathbf{\Sigma}$. As a consequence, the solution will not be ill-conditioned when $n < p$ (HD-LSS underdetermined case). In the Gaussian case, $\mathbf{z}_i \sim \mathbb{C}N_p(\mathbf{0}, \mathbf{\Sigma})$, we have that $\rho(t) = t$ and the solution to (6) is easily shown to be $\hat{\mathbf{\Sigma}} = \mathbf{S} + \lambda \mathbf{I}$, where $\mathbf{S} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{z}_i \mathbf{z}_i^{\mathrm{H}}$ denotes the SCM. We shall consider generalization of the methodology in [1, 7] and consider shrinkage $M$-estimators of $\mathbf{\Sigma}$ for general $\rho$ functions.

*Applications:* Conventional beamforming cannot be used in HD-LSS underdetermined problems, e.g., the minimum variance distortionless response (MVDR) beamformer weight vector requires the inverse of the SCM. Applying the diagonal loading, i.e., using $\mathbf{S} + \lambda \mathbf{I}$ in place of $\mathbf{S}$ is the commonly used approach; see [3]. Our simulations show that robust shrinkage covariance matrix estimators (e.g., shrinkage Tyler's $M$-estimator [1, 7] or Huber's $M$-estimator proposed here) provide superior performance in non-Gaussian impulsive noise.

## References


[1] Y. Chen, A. Wiesel, and A. O. Hero. Robust shrinkage estimation of high-dimensional covariance matrices. *IEEE Trans. Signal Processing*, 59(9):4097 – 4107, 2011.

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

[3] J. Li, P. Stoica, and Z. Wang. On robust Capon beamforming and diagonal loading. *IEEE Trans. Signal Processing*, 51(7):1702 – 1715, 2003.

[4] R. A. Maronna. Robust ridge regression for high-dimensional data. *Technometrics*, 53(1):44–53, 2011.

[5] E. Ollila, D. E. Tyler, V. Koivunen, and H. V. Poor. Complex elliptically symmetric distributions: survey, new results and applications. *IEEE Trans. Signal Processing*, 60(11):5597 – 5625, 2012.

[6] S. A. Razavi, E. Ollila, and V. Koivunen. Robust greedy algorithms for compressed sensing. In *Proc. 20th European Signal Processing Conference (EUSIPCO'12)*, pages 969–973, Bucharest, Romania, 2012.

[7] A. Wiesel. Geodesic convexity and covariance estimation. *IEEE Trans. Signal Processing*, 60(12):6182–6189, 2012.

# Robust Near-Separable Nonnegative Matrix Factorization Using Linear Optimization

Nicolas Gillis
ICTEAM Institute
Université catholique de Louvain
B-1348 Louvain-la-Neuve
Email: nicolas.gillis@uclouvain.be

Robert Luce
Institut für Mathematik
Technische Universität Berlin
Strasse des 17. Juni 136 - 10623 Berlin, Germany
Email: luce@math.tu-berlin.de

**Abstract**: Nonnegative matrix factorization (NMF) has been shown recently to be tractable under the separability assumption, which amounts for the columns of the input data matrix to belong to the convex cone generated by a small number of columns. Bittorf, Recht, Ré and Tropp ('Factoring nonnegative matrices with linear programs', NIPS 2012) proposed a linear programming (LP) model, referred to as HottTopixx, which is robust under any small perturbation of the input matrix. However, HottTopixx has two important drawbacks: (i) the input matrix has to be normalized, and (ii) the factorization rank has to be known in advance. In this talk, we generalize HottTopixx in order to resolve these two drawbacks, that is, we propose a new LP model which does not require normalization and detects the factorization rank automatically. Moreover, the new LP model is more flexible, significantly more tolerant to noise, and can easily be adapted to handle outliers and other noise models. We show on several synthetic datasets that it outperforms HottTopixx while competing favorably with two state-of-the-art methods.

**Keywords**: nonnegative matrix factorization, linear programming, robustness to noise

## 1 Introduction

Nonnegative matrix factorization (NMF) is a powerful dimensionality reduction technique as it automatically extracts sparse and meaningful features from a set of nonnegative data vectors. Given $n$ nonnegative $m$-dimensional vectors gathered in a nonnegative matrix $M \in \mathbb{R}^{m\times n}_+$ and a factorization rank $r$, NMF computes two nonnegative matrices $W \in \mathbb{R}^{m\times r}_+$ and $H \in \mathbb{R}^{r\times n}_+$ such that $M \approx WH$. Unfortunately, NMF is NP-hard in general [8]. However, if the input data matrix $M$ is $r$-separable, that is, if it can be written as $M = W\,[I_r,\, H']\Pi$, where $I_r$ is the $r$-by-$r$ identity matrix, $H' \geq 0$ and $\Pi$ is a permutation matrix, then the problem can be solved in polynomial time [2]. Separability means that there exists an NMF $(W, H) \geq 0$ of $M$ of rank $r$ where each column of $W$ is equal to a columns of $M$. Geometrically, $r$-separability means that the cone generated by the columns of $M$ has $r$ extreme rays given by the columns of $W$. Equivalently, if the columns of $M$ are normalized to sum to one, $r$-separability means that the convex hull generated by the columns of $M$ has $r$ vertices given by the columns of $W$; but see, e.g., [7]. The separability assumption makes sense in several applications, e.g., text mining, hyperspectral unmixing and blind source separation [6]. Several algorithms have been proposed to solve the near-separable NMF problem, e.g., [2] [6] [7], which refers to the NMF problem of a separable matrix $M$ to which some noise is added; see Section 2. In this talk, our focus is on the LP model proposed by Bittorf, Recht, Ré and Tropp [3] and referred to as HottTopixx. It is described in the next section.

## 2 HottTopixx

A matrix $M$ is $r$-separable if and only if

$$M = WH = W[I_r, H']\Pi = [W, WH']\Pi$$
$$= [W, WH']\Pi \underbrace{\Pi^{-1} \begin{pmatrix} I_r & H' \\ 0_{(n-r)\times r} & 0_{(n-r)\times(n-r)} \end{pmatrix} \Pi}_{X^0 \in \mathbb{R}^{n\times n}_+},$$

for some permutation $\Pi$ and some matrix $H' \geq 0$. The matrix $X^0$ is an $n$-by-$n$ nonnegative matrix with $(n-r)$ zero rows such that $M = MX^0$. Assuming the columns of $M$ sum to one, the columns of $W$ and $H'$ have sum to one as well. Based on these observations, Bittorf, Recht, Ré and Tropp [3] proposed to solve the following optimization problem in order to identifying approximately the columns of the matrix $W$ among the columns of the noisy separable matrix $\tilde{M} = WH + N$

with $||N||_1 = \max_j ||N(:,j)||_1 \leq \epsilon$ :

$$\begin{aligned} \min_{X \in \mathbb{R}_+^{n \times n}} \quad & p^T \operatorname{diag}(X) \\ \text{such that} \quad & ||\tilde{M} - \tilde{M}X||_1 \leq 2\epsilon, \\ & \operatorname{tr}(X) = r, \\ & X(i,i) \leq 1 \text{ for all } i, \\ & X(i,j) \leq X(i,i) \text{ for all } i,j, \end{aligned} \tag{1}$$

where $p$ is any $n$-dimensional vector with distinct entries. The $r$ largest diagonal entries of an optimal solution $X^*$ of (1) correspond to columns of $\tilde{M}$ close to the columns of $W$, given that the noise is sufficiently small [4]. However, it has two important drawbacks:

- the factorization rank $r$ has to be chosen in advance so that the LP above has to be resolved when it is modified (in fact, in practice, a 'good' factorization rank for the application at hand is typically found by a trial-and-error approach),
- the columns of the input data matrix have to be normalized in order to sum to one. This may introduce important distortions in the dataset and lead to poor performances [7].

## 3 Contribution

In this talk, we generalize HottTopixx in order to resolve the two drawbacks mentioned above. More precisely, we propose a new LP model which has the following properties:

- It detects the number $r$ of columns of $W$ automatically.
- It can be adapted to dealing with outliers.
- It does not require column normalization.
- It is significantly more tolerant to noise than HottTopixx. In fact, we propose a tight robustness analysis of the new LP model proving its superiority.

This is illustrated on several synthetic datasets, where the new LP model is shown to outperform HottTopixx while competing favorably with two state-of-the-art methods, namely the successive projection algorithm (SPA) from [1, 6] and the fast conical hull algorithm (XRAY) from [7]; see Figure 1 for an illustration on synthetic datasets. We refer the reader to [5] for all the details about the proposed algorithm, including the proof of robustness and more numerical experiments.

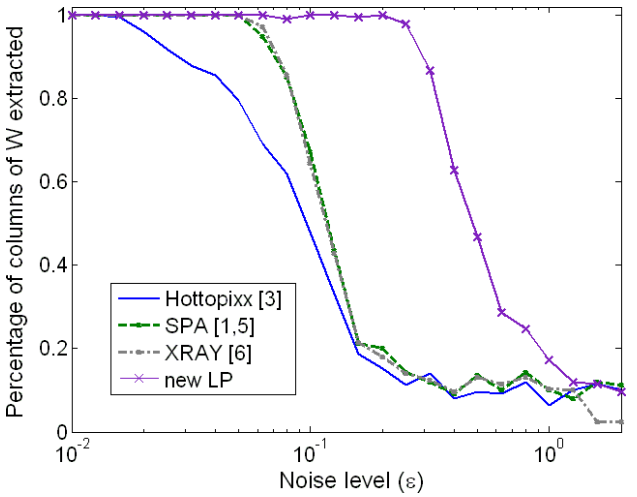



**Fig. 1:** Comparison of near-separable NMF algorithms on synthetic datasets. The noisy separable matrices are generated as follows: each entry of $W \in \mathbb{R}_+^{50 \times 100}$ is generated uniformly at random in $[0,1]$ and then each column of $W$ is normalized to sum to one, each column of $H'$ in $\mathbb{R}_+^{10}$ is generated using a Dirichlet distribution whose parameters are picked uniformly at random in $[0,1]$, and the noise $N$ contains one non-zero entry in each column so that $||N||_1 = \epsilon$. For each noise level $\epsilon$, we generate 25 such matrices and display the percentage of columns of $W$ correctly extracted by the different algorithms (hence the higher the curve, the better).

## References


[1] U. Araújo, B. Saldanha, R. Galvão, T. Yoneyama, H. Chame, and V. Visani, "The successive projections algorithm for variable selection in spectroscopic multicomponent analysis," *Chemom. and Intell. Lab. Syst.*, vol. 57, no. 2, pp. 65–73, 2001.

[2] S. Arora, R. Ge, R. Kannan, and A. Moitra, "Computing a nonnegative matrix factorization – provably," in *STOC '12*, 2012, pp. 145–162.

[3] V. Bittorf, B. Recht, E. Ré, and J. Tropp, "Factoring nonnegative matrices with linear programs," in *NIPS' 12*, 2012, pp. 1223–1231.

[4] N. Gillis, "Robustness analysis of hotttopixx, a linear programming model for factoring nonnegative matrices," 2012, arXiv:1211.6687.

[5] N. Gillis and R. Luce, "Robust near-separable nonnegative matrix factorization using linear optimization," 2013, arXiv:1302.4385.

[6] N. Gillis and S. Vavasis, "Fast and robust recursive algorithms for separable nonnegative matrix factorization," 2012, arXiv:1208.1237.

[7] A. Kumar, V. Sindhwani, and P. Kambadur, "Fast conical hull algorithms for near-separable non-negative matrix factorization," in *International Conference on Machine Learning (ICML)*, 2013.

[8] S. Vavasis, "On the complexity of nonnegative matrix factorization," *SIAM J. on Optimization*, vol. 20, no. 3, pp. 1364–1377, 2009.

# Data-Driven and Problem-Oriented Multiple-Kernel Learning

Valeriya Naumova
Johann Radon Institute for Computational and Applied Mathematics
Altenbergerstrasse 69, 4040 Linz - Austria
valeriya.naumova@oeaw.ac.at

Sergei V. Pereverzyev
Johann Radon Institute for Computational and Applied Mathematics
Altenbergerstrasse 69, 4040 Linz - Austria
sergei.pereverzyev@oeaw.ac.at

**Abstract**: The paper is concerned with an adaptive kernel design for addressing the problem of the function reconstruction inside/outside the scope of given data points. We analyze the state of the art methods and, by doing that, we show a need for a novel and a more sophisticated approach to a data-driven and problem-oriented kernel design. Finally, we present such approach and show its superiority with respect to the known methods on the numerical experiments with real data.

**Keywords**: reproducing kernel Hilbert space, multiple kernel design, approximation theory

## 1 Introduction

In recent years there has been a very fast growing interest in defining and analyzing mathematical predictive models, that reconstruct/predict a real-valued function $f$ defined on $X \subset \mathbb{R}^d$ from available noisy data $\mathbf{z} = \{(x_i, y_i)\} \subset X \times \mathbb{R}$, where $y_i = f(x_i) + \xi_i$, and $\xi_i$ is a measurement error.

It is a well-known fact that such reconstruction problem is ill-posed [2], and its numerical treatment can be performed in a stable way only by applying special regularization methods.

The most popular among them is the Tikhonov method, which in the present context consists in constructing the approximant for $f$ as the minimizer $f_\lambda$ of the functional

$$T_\lambda(f; \mathcal{H}, \mathbf{z}) := \frac{1}{|\mathbf{z}|} \sum_{i=1}^{|\mathbf{z}|} (f(x_i) - y_i)^2 + \lambda \|f\|_{\mathcal{H}}^2, \qquad (1)$$

where $|\mathbf{z}|$ is the cardinality of the set $\mathbf{z}$ and $\lambda$ is a regularization parameter, which trades-off data error with smoothness measured in terms of a space $\mathcal{H}$.

At the same time, the efficient application of this scheme is governed by two issues: one of them is the choice of the regularization parameter $\lambda$, which has been extensively studied within the classical regularization theory, and another one, which is a challenging and a central problem [4], is the choice of a space $\mathcal{H}$, whose norm is used for penalization.

However, a proper choice of a suitable space $\mathcal{H}$ has so far been elusive, since in most known studies the norm of the penalty term in (1) was assumed to be given a priori, for instance, by norm of some Sobolev space [9, 10]. At the same time, keeping in mind that a Sobolev space is a particular example of a *Reproducing Kernel Hilbert Space (RKHS)* $\mathcal{H}_K$, induced by a symmetric positive definite function $K(t, x)$, $t, x \in X$, called a kernel, the issue about the choice of a proper regularization space is, in fact, about the choice of a kernel for an RKHS.

One of the leading concepts behind the clarification of this issue in the past few years has been universal kernels [5], which potentially allow to construct $f_\lambda$ having a good approximating property. However, as it can be seen from numerical experiments in [6] the concept of the kernel's universality does not guarantee good approximation property at points outside the scope of seen inputs, which is our main interest. The same is true for the kernels given as radial basis functions. Therefore, for prediction of the unknown functional dependency outside the scope of seen inputs, the question about a proper choice of a regularization kernel is, in general, open until now. In this talk we aim to shed light on this important but as of yet under-researched problem. Moreover, we are going to show how the clarification of this issue could help to improve management of diabetes, namely by providing more accurate prediction of future blood glucose concentration.

## 2 Data-driven and problem-oriented kernel design

Lanckriet et al. [3] were among the first to emphasize the need to consider multiple kernels or parameterizations of kernels, and not a single a priori fixed kernel. These authors advocate the approach to a data-driven kernel choice, where one tries to find a "good" kernel $K$ as a linear combination

$$K = \sum_{j=1}^{N} \beta_j K_j \tag{2}$$

of a priori prescribed kernels $K_j \in \mathcal{K}(\{K_j\})$, $j = 1, 2, \ldots, N$. Such approach, often referred to as the multiple kernel learning (MKL), has been an attractive topic in learning theory (see, e.g., [8] and references therein). At the same time, most MKL-methods employ a standard simplex constraint $\beta_1 + \beta_2 + \ldots + \beta_N = 1,\ \beta_j > 0,\ j = 1, 2, \ldots, N$, on the combination weights of kernels (2). As a result, the minimization of the Tikhonov functional (1) in this case is performed again with a priori given regularization term that means that most MKL-methods are not really intended for a data-driven choice of the regularization space.

Moreover, it is worthy of notice that for some practical applications the set of linear combinations of kernels $\mathcal{K}(\{K_j\})$ is not rich enough, and, thus, more general parameterizations are also of interest.

Let us consider a set $\mathcal{K}(X)$ of all kernels defined on $X \subset \mathbb{R}^d$. Let also $\Omega$ be a compact metric space and $G : \Omega \to \mathcal{K}(X)$ be an injection such that for any $t, x \in X$, the function $\omega \to G(\omega)(t, x)$ is a continuous map from $\Omega$ to $\mathbb{R}$, here $G(\omega)(t, x)$ is the value of the kernel $G(\omega) \in \mathcal{K}(X)$ at $(t, x) \in X \times X$.

Each such mapping $G$ determines a set of kernels

$$\mathcal{K}(\Omega, G) = \{K : K = G(\omega), K \in \mathcal{K}(X), w \in \Omega\}$$

parameterized by elements of $\Omega$. In contrast to the set of linear combinations of kernels, $\mathcal{K}(\Omega, G)$ may be a non-linear manifold.

Assume now that one is interested in choosing a kernel $K$ from some admissible set $\mathcal{K}(\Omega, G)$ of parameter-dependent kernels $K = K(\omega; t, x)$, where all parametric dependences are collected in a parameter vector $\omega \in \Omega \subset \mathbb{R}^l$.

To this end, we present a novel approach to the data-driven kernel choice from $\mathcal{K}(\Omega, G)$ based on the concept of meta-learning [1, 7]. In particular, the kernel is learned to adjust to each given input data $x$ by reconstructing a vector function $\omega = \omega(x)$ governing the choice of the kernel $K = K(\omega; t, x)$, from $\mathcal{K}(\Omega, G)$ on the basis of experience with similar tasks. In addition, we illustrate that the presented approach is superior to the "traditional" kernel selection procedure, in which a kernel is chosen "globally" for the whole given training set $\mathbf{z}$, but it does not account for particular features of input $x$. The constructed data-driven approach yields better performance as it is seen from the results of the extensive numerical experiments with real clinical data for blood glucose prediction problem [6, 7].

We argue that the presented approach is a new promising direction that could lead to the construction of the efficient data-driven algorithms. It appears that such approach has not been systematically studied so far in the framework of the regularization theory. The investigations presented in the current work are contributing to the first steps in this challenging direction.

## References


[1] P. Brazdil, C. Giraud-Carrier, C. Soares, and R. Vilalta. *Metalearning: Applications to Data Mining*. Springer-Verlag, Berlin Heidelberg, 2009.

[2] H. Engl, M. Hanke, and A. Neubauer. *Regularization of Inverse Problems*, volume 375 of *Mathematics and Its Applications*. Kluwer Academic Publishers, Dordrecht, Boston, London, 1996.

[3] G. Lanckriet, N. Christianini, L. Ghaoui, P. Bartlett, and M. Jordan. Learning the kernel matrix with semidefinite programming. *J. Mach. Learn. Res.*, 5:27–72, 2004.

[4] C. A. Micchelli and M. Pontil. Learning the kernel function via regularization. *J. Mach. Learn. Res.*, 6:1099–1125, 2005.

[5] C. A. Micchelli, Y. Xu, and H. Zhang. Universal kernels. *J. Mach. Learn. Res.*, 7:2651–2667, 2006.

[6] V. Naumova, S. V. Pereverzyev, and S. Sampath. Extrapolation in variable RKHSs with application to the blood glucose reading. *Inverse Problems*, 27(7):075010. 13 pp., 2011.

[7] V. Naumova, S. V. Pereverzyev, and S. Sivananthan. A meta-learning approach to the regularized learning – case study: Blood Glucose prediction. *Neural Networks*, 33:181–193, 2012.

[8] C. S. Ong, A. Smola, and R. Williamson. Learning the kernel with hyperkernels. *J. Machine Learning Research*, 6:1043–1071, 2005.

[9] A. N. Tikhonov and V. B. Glasko. *Use of the Regularization Methods in Non-Linear Problems*, volume 5. USSR Comput. Math. Phys., 1965.

[10] G. Wahba. *Spline Models for Observational Data*, volume 59 of *Series in Applied Mathematics*. CBMS-NSF Regional Conf., SIAM, 1990.

# Support Vector Machine with spatial regularization for pixel classification

R. Flamary
Lagrange Lab., Université de Nice Sophia-Antipolis,
CNRS, Observatoire de la Côte d'Azur, Nice, France
remi.flamary@unice.fr

A. Rakotomamonjy
LITIS Lab.
Université de Rouen, France
alain.rakoto@insa-rouen.fr

**Abstract**: We propose in this work to regularize the output of a svm classifier on pixels in order to promote smoothness in the predicted image. The learning problem can be cast as a semi-supervised SVM with a particular structure encoding pixel neighborhood in the regularization graph. We provide several optimization schemes in order to solve the problem for linear SVM with $\ell_2$ or $\ell_1$ regularization and show the interest of the approach on an image classification example with very few labeled pixels.

**Keywords**: Support Vector Machine, Large scale learning, semi-supervised learning

## 1 Introduction

Pixel classification is the problem of assigning a class to every pixel in an image. This is a classical problem with several applications in medical imaging or in geoscience remote sensing where it is denoted as image classification [2, 6]. A common approach for solving this problem is to use discriminative machine learning techniques and to treat pixels as independent vectors. In order to take into account the spatial prior over the pixels, several approaches have been proposed. One example is to include spatial features or kernels to the pixel representation such as filter output images [6]. Another approach is to use a post-processing on the output of the classifier, for instance by using a Markov Random Field to include spatial information [8]. While the post-processing approach can integrate high order relations between pixels, it is also more computationally intensive.

Another challenge of pixel classification is the dataset itself. The number of pixel $N$ increases quadratically with the size of the image and the number $n$ of labeled pixels is usually small. This suggest the use of semi-supervised learning methods [2] which have led to dramatic performance improvement when the number of labeled pixels is small. Note that in their works, the large number of pixels are handled using low-rank kernel approximations, leading to the learning of a linear SVM on a small number $d$ of nonlinear features.

In this paper, we focus on linear SVM applied on $d \ll N$ features (potentially nonlinear) extracted form the data. We want not only to use unlabeled pixels in the learning problem but also to promote spatial smoothness on the output of the prediction function, thus using unlabeled pixels. We propose to this end to regularize the SVM output using a term that encodes the spatial neighborhood of the pixels as seen in [5]. This approach is a particular case of manifold-based regularized semi-supervised learning. We discuss in the following how to solve the learning problem for different regularizations on the linear SVM. Finally, numerical experiments are performed in order to show the interest of the approach on a difficult pixel classification problem.

## 2 SVM with spatial regularization

The dataset consists in a full image of $N$ pixels with $d$ features per pixel (possibly hyperspectral spectrum or other features). These pixels $\mathbf{x}$ are stored in the matrix $\mathbf{X} \in \mathbb{R}^{N \times d}$. Only $n < N$ of these pixels with indexes $i \in \mathcal{L}$ are labeled with $y_i \in \{-1, +1\}$. We want to learn a prediction function $f(\cdot)$ of the form

$$f(\mathbf{x}) = \sum_i w_i x_i + b = \mathbf{w}^\top \mathbf{x} + b \tag{1}$$

where $\mathbf{w} \in \mathbb{R}^d$ is the normal vector to the separating hyperplane and $b \in \mathbb{R}$ is a bias term.

### 2.1 Learning problem

We propose to learn the prediction function with the following optimization problem:

$$\min_f \sum_{i \in \mathcal{L}} H(y_i, f(\mathbf{x}_i)) + \lambda_s \sum_{i,j} W_{i,j} (f(\mathbf{x}_i) - f(\mathbf{x}_j))^2 + \lambda_r \Omega(f) \tag{2}$$

where $H(y, f(\mathbf{x})) = \max(0, 1 - yf(\mathbf{x}))^2$ is the squared hinge loss, $\mathbf{W} \in \mathbb{R}^{N \times N}$ is a symmetric matrix of general term $W_{i,j}$ that encodes the similarity between pixel $i$ and $j$, $\lambda_s$ and $\lambda_r$ are regularization parameters and $\Omega(\cdot)$ is the SVM regularization term.

This problem is a classical semi-supervised learning problem. If the similarity matrix $\mathbf{W}$ were chosen to be a Gaussian kernel matrix, the problem would boil down

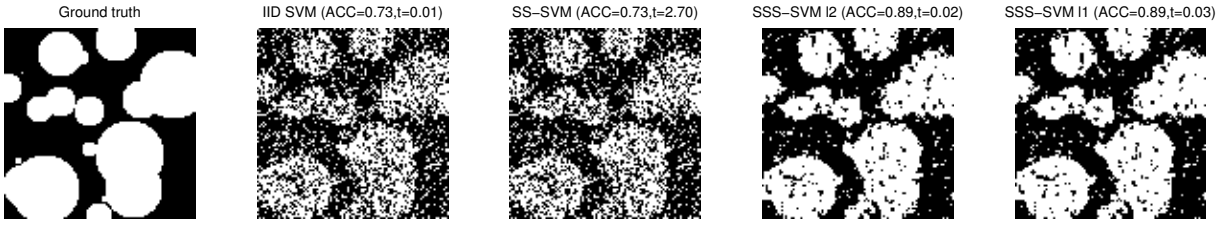



**Fig. 1:** Ground truth labels, accuracy ACC, training time t in second and decision maps for IID SVM, SS-SVM and SSS-SVM with $\ell_2$ and $\ell_1$ regularization. The regularization parameters of each methods are selected in order to maximize test accuracy.

to a Laplacian SVM [2]. But it requires the computation of a $\mathcal{O}(N^2)$ kernel matrix. In our case, we want to promote smoothness in the output on the prediction function *i.e.* we want neighbor pixels to have similar prediction score. To this end, we propose a $\mathbf{W}$ matrix such that $W_{i,j} = 0$ everywhere except when pixels $\mathbf{x}_i$ and $\mathbf{x}_j$ are spatial neighbors ($W_{i,j} = 1$). Note that this regularization is similar to a total variation regularization but with a quadratic penalty term. Moreover the Laplacian regularization term can be computed with a complexity $\mathcal{O}(N)$ which is essential to large scale learning.

### 2.2 Optimization algorithm

Problem (2) can be reformulated in the linear case as

$$\min_{\mathbf{w},b} \sum_{i \in \mathcal{L}} H(y_i, \mathbf{w}^\top \mathbf{x}_i + b) + \lambda_s \mathbf{w}^\top \mathbf{\Sigma} \mathbf{w} + \lambda_r \Omega(\mathbf{w}) \quad (3)$$

where $\mathbf{\Sigma} = \mathbf{X}^\top(\mathbf{D} - \mathbf{W})\mathbf{X}$ with $\mathbf{D}$ the diagonal matrix such that $D_{i,i} = \sum_{j=1}^{N} W_{i,j}$.

When $\Omega(\mathbf{w}) = \|\mathbf{w}\|_2^2$, the problem is a classical $\ell_2$ SVM with a metric regularization ($\widetilde{\mathbf{\Sigma}} = \mathbf{\Sigma} + \lambda_r/\lambda_s \mathbf{I}$). One approach suggested by [7] and [5] is to perform a change of variable $\tilde{\mathbf{w}} = \mathbf{\Sigma}^{1/2}\mathbf{w}$ and $\tilde{\mathbf{x}} = \mathbf{\Sigma}^{-1/2}\mathbf{x}$. The resulting problem can be solved with a classical linear SVM solver such as the one proposed by [3].

When $\Omega(\cdot)$ is a more complex regularization term such as the $\ell_1$ norm, we propose to use a proximal splitting algorithm such as ADMM to solve the problem [1]. This approach allows us to use iteratively the efficient solver discussed above while integrating prior information to the problem through regularization.

## 3 Numerical experiments

Numerical experiments are performed on a simulated image of size $(100 \times 100)$. The simulated image that can be seen in Fig. 1 is generated as follows: i) the ground truth image is obtained by generating random circles in the image that are set to $+1$ ($-1$ for the background), ii) 10 discriminant features are generated by applying Gaussian noise to the ground truth image ($\sigma = 5$), iii) the previous images are filtered by a $3 \times 3$ average filter and a $3 \times 3$ median filter resulting in 20 additional features. iv) 10 images containing only Gaussian noise are added to obtain 40 features.

In order to demonstrate the interest of our approach we randomly select 10 labeled samples from each classes and we learn an independent SVM (IID-SVM), a semi-supervised Laplacian SVM (SS-SVM), and our proposed approach, the spatially regularized semi-supervised SVM (SSS-SVM) for both $\ell_2$ and $\ell_1$ regularization. Results show that smooth classification and prediction maps are enforced leading to an important improvement in recognition performances (see Fig. 1).

## References

[1] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

[2] G. Camps-Valls, T. Bandos Marsheva, and D. Zhou. Semi-supervised graph-based hyperspectral image classification. *Geoscience and Remote Sensing, IEEE Transactions on*, 45(10):3044–3054, 2007.

[3] O. Chapelle. Training a support vector machine in the primal. *Neural Comput.*, 19(5):1155–1178, 2007.

[4] M. Dundar, B. Krishnapuram, J. Bi, and R.B. Rao. Learning classifiers when the training data is not iid. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, 2007.

[5] M. Dundar, J. Theiler, and S. Perkins. Incorporating spatial contiguity into the design of a support vector machine classifier. In *IGARSS*, pages 364–367. IEEE, 2006.

[6] M. Fauvel, J. Chanussot, and J. A. Benediktsson. A spatial-spectral kernel-based approach for the classification of remote-sensing images. *Pattern Recogn.*, 45:381–392, January 2012.

[7] V. Sindhwani, P. Niyogi, M. Belkin, and S. Keerthi. Linear manifold regularization for large scale semi-supervised learning. In *Proc. of the 22nd ICML Workshop on Learning with Partially Classified Training Data*, 2005.

[8] Y. Tarabalka, M. Fauvel, J. Chanussot, and J.A. Benediktsson. Svm-and mrf-based method for accurate classification of hyperspectral images. *Geoscience and Remote Sensing Letters, IEEE*, 7(4):736–740, 2010.

# Regularized structured low-rank approximation

Mariya Ishteva, Konstantin Usevich, Ivan Markovsky
Dept. ELEC, Vrije Universiteit Brussel
{mariya.ishteva, konstantin.usevich, ivan.markovsky}@vub.ac.be

**Abstract**: We consider the problem of approximating a linearly structured matrix, for example a Hankel matrix, by a low-rank matrix with the same structure. This problem occurs in system identification, signal processing and computer algebra, among others. We impose the low-rank by modeling the approximation as a product of two factors with reduced dimension. The structure of the low-rank model is enforced by introducing a regularization term in the objective function. In contrast to approaches based on kernel representations (in linear algebraic sense), the proposed algorithm is designed to address the case of small targeted rank.

**Keywords**: low-rank approximation, affine structure, regularization

## 1 Introduction

Low-rank approximations are widely used in data mining, machine learning and signal processing as a tool for dimensionality reduction and factor analysis. In system identification, signal processing and computer algebra, the matrices are often structured, e.g., (block) Hankel, (block) Toeplitz, Sylvester, or banded matrices with fixed bandwidth. Note that sparse matrices are also structured matrices. Unstructured matrices can be considered as a special case of structured matrices as well. Therefore, the goal of structured low-rank approximation is to preserve the given structure while obtaining a low-rank approximation.

Existing approaches [1–4, 6] are based on kernel representation or on solving a system of linear equations in the total least squares sense. We propose a new view of the problem that makes connection with the machine learning literature.

We consider an image representation of the approximation, i.e., given a structured matrix $D \in \mathbb{R}^{m\times n}$ and a number $r$ such that $r \ll m, n$, find two factors $P \in \mathbb{R}^{m\times r}$ and $L \in \mathbb{R}^{r\times n}$, such that

$$D \approx PL \quad \text{and} \quad PL \text{ is a structured matrix.}$$

Although each of the constraints can easily be handled separately, imposing both low-rank and fixed structure on the approximation is nontrivial.

In contrast to kernel representations which are more efficient for large $r$, the proposed approach is meant for problems with small $r$. Finally, for general structures, existing kernel approaches have restrictions on the possible values of the rank $r$. With the new approach we can overcome this limitation.

## 2 Problem formulation

### 2.1 Structures

Affine structures can be defined as

$$\mathcal{S}(p) = S_0 + \sum_{k=1}^{n_p} S_k p_k,$$

where $S_0, S_1, \ldots, S_{n_p} \in \mathbb{R}^{m\times n}$, $p \in \mathbb{R}^{n_p}$ and $n_p \in \mathbb{N}$ is the (minimal) number of parameters. Let $\operatorname{vec}(X)$ denote the vectorized matrix $X$ and let

$$\mathbf{S} = \begin{bmatrix}\operatorname{vec}(S_1) & \cdots & \operatorname{vec}(S_{n_p})\end{bmatrix} \in \mathbb{R}^{mn\times n_p}.$$

Since $n_p$ is minimal, $\mathbf{S}$ has full column rank.

For simplicity, we assume that $S_0 = \mathbf{0}$, the elements of $\mathbf{S}$ are only 0 and 1, and there is at most one nonzero in each row (non-overlap across $S_k$), i.e., every element of the structured matrix corresponds to only one element of $p$.

### 2.2 Orthogonal projection on image($\mathcal{S}$)

Let $\Pi_{\mathbf{S}} = (\mathbf{S}^\top \mathbf{S})^{-1}\mathbf{S}^\top$. It can be shown that the orthogonal projection of a matrix $X$ on image($\mathcal{S}$) is given by

$$P_{\mathcal{S}}(X) \equiv \mathcal{S}(\Pi_{\mathbf{S}} \operatorname{vec}(X)). \tag{1}$$

The effect of applying $\Pi_{\mathbf{S}}$ on a vectorized matrix $X$ is producing a $p_X$ structure vector by averaging elements corresponding to the same $S_k$. Note that applying $\Pi_{\mathbf{S}}$ on a (vectorized) structured matrix extracts its structure vector, since $\Pi_{\mathbf{S}}\,\mathbf{S}\,p = p$. Finally,

$$\operatorname{vec}(P_{\mathcal{S}}(X)) = \mathbf{S}\,\Pi_{\mathbf{S}} \operatorname{vec}(X).$$

### 2.3 Optimization problem

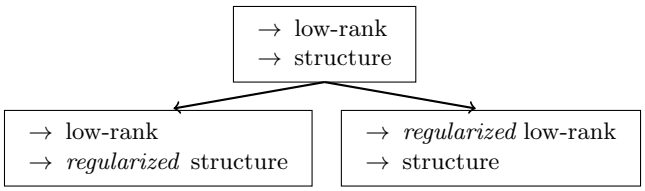
The weighted structured low-rank approximation problem is formulated as

$$\min_{\hat{p}} \|p - \hat{p}\|_{\overline{W}}, \quad \text{such that} \quad \operatorname{rank}(\mathcal{S}(\hat{p})) \leq r, \qquad (2)$$

where $\overline{W} \in \mathbb{R}^{n_p \times n_p}$ is a symmetric positive definite matrix of weights and $\|x\|^2_{\overline{W}} = x^\top \overline{W}\, x$. If $\overline{W}$ is the identity matrix, $\|\cdot\|_{\overline{W}} = \|\cdot\|_2$.

Although each of the constraints can easily be handled separately, imposing both low-rank and fixed structure on the approximation is nontrivial. We approach the problem with a regularization technique.

We have the following two choices:




- Regularize the structure constraint

$$\min_{P,L} \|D - PL\|^2_W + \lambda \|PL - P_{\mathcal{S}}(PL)\|^2_F, \qquad (3)$$

where $W$ is a weight matrix, $\lambda$ is a regularization parameter and $F$ stands for the Frobenius norm.

- Regularize the rank constraint

$$\min_{P,L} \|D - P_{\mathcal{S}}(PL)\|^2_W + \lambda \|PL - P_{\mathcal{S}}(PL)\|^2_F. \qquad (4)$$

Note that for $\lambda = \infty$ the term $\|PL - P_{\mathcal{S}}(PL)\|$ have to be 0 and the three problems (2), (3) and (4) are equivalent. The interpretations of (3) and (4) are however different. In (4) the main part is the structure and the low rank is 'secondary'. In (3) it is the other way around, although in both cases both constraints are satisfied at the solution. We will focus on (4), since it can be formulated using $\overline{W}$ in the following way

$$\min_{P,L} \|p - \Pi_{\mathbf{S}} \operatorname{vec}(PL)\|^2_{\overline{W}} + \lambda \|PL - P_{\mathcal{S}}(PL)\|^2_F.$$

## 3 The proposed algorithm

### 3.1 High-level idea

We solve the minimization problem (4) by alternatingly improving the approximations of $P$ and of $L$

$$\begin{aligned} &\min_{L} \|p - \Pi_{\mathbf{S}} \operatorname{vec}(PL)\|^2_{\overline{W}} + \lambda \|PL - P_{\mathcal{S}}(PL)\|^2_F, \\ &\min_{P} \|p - \Pi_{\mathbf{S}} \operatorname{vec}(PL)\|^2_{\overline{W}} + \lambda \|PL - P_{\mathcal{S}}(PL)\|^2_F \end{aligned} \qquad (5)$$

until convergence.

### 3.2 Details

Let $I_n$ be the $n \times n$ identity matrix, '$\otimes$' denote the Kronecker product and $\overline{W} = \overline{M}^\top \overline{M}$. Using (1) and the equality $\operatorname{vec}(XYZ) = (Z^\top \otimes X) \operatorname{vec}(Y)$, (5) can be reformulated as

$$\min_{L} \left\| \begin{bmatrix} \overline{M}\, \Pi_{\mathbf{S}} \\ \sqrt{\lambda}(I_{mn} - \mathbf{S}\, \Pi_{\mathbf{S}}) \end{bmatrix} (I_n \otimes P) \operatorname{vec}(L) - \begin{bmatrix} \overline{M}p \\ \mathbf{0} \end{bmatrix} \right\|^2_2,$$

$$\min_{P} \left\| \begin{bmatrix} \overline{M}\, \Pi_{\mathbf{S}} \\ \sqrt{\lambda}(I_{mn} - \mathbf{S}\, \Pi_{\mathbf{S}}) \end{bmatrix} (L^\top \otimes I_m) \operatorname{vec}(P) - \begin{bmatrix} \overline{M}p \\ \mathbf{0} \end{bmatrix} \right\|^2_2.$$

These are least squares problems and can easily be solved by standard techniques.

The matrix $P$ can be initialized by a matrix representing the left dominant subspace of $A$. We declare that $PL$ is a structured matrix if

$$\|PL - P_{\mathcal{S}}(PL)\|^2_F < 10^{-12}.$$

### 3.3 Parameter $\lambda$

In theory, if we fix $\lambda = \infty$, then we have the exact structured low-rank approximation problem. In practice, we start from a small value and increase it with each iteration until it reaches a "large enough" value. This way we allow the algorithm to move to a "good region" quickly and then impose more strictly all constraints. For convergence properties, we rely on the theory of quadratic penalty method from [5, §17.1].

## References

[1] M. Chu and G. H. Golub. *Inverse eigenvalue problems*. Oxford University Press, 2005.

[2] B. De Moor. Structured total least squares and $L_2$ approximation problems. *Linear Algebra Appl.*, 188–189:163–207, 1993.

[3] I. Markovsky. Structured low-rank approximation and its applications. *Automatica*, 44(4):891–909, 2008.

[4] I. Markovsky and K. Usevich. Software for weighted structured low-rank approximation. Technical Report 339974, Univ. of Southampton, 2012.

[5] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer Series in Operations Research. Springer, 2nd edition, 2006.

[6] K. Usevich and I. Markovsky. Variable projection for affinely structured low-rank approximation in weighted 2-norm, 2012. Available from `http://arxiv.org/abs/1211.3938`.

# A Heuristic Approach to Model Selection for Online Support Vector Machines

Davide Anguita, Alessandro Ghio, Isah Abdullahi Lawal, Luca Oneto
DITEN - University of Genoa, Via Opera Pia 11A, I-16145, Genoa, Italy
{Davide.Anguita, Alessandro.Ghio, Abdullahi.LawalIsah, Luca.Oneto}@unige.it

**Abstract**: In this abstract, we cope with the problem of Model Selection (MS) of Support Vector Machine (SVM) classifiers in the Online Learning (OL) framework. Though often neglected in OL, MS is paramount to guarantee that model effectiveness is guaranteed while exploiting new gathered samples. In particular we propose a heuristic approach, which can be feasibly applied in OL applications to SVM. The effectiveness of the proposal is supported by preliminary experimental results.

**Keywords**: Online Learning, Support Vector Machines, Model Selection

## 1 Introduction

Support Vector Machines (SVMs) [9, 10] are one of the most effective technique for classification purposes. The SVMs learning phase is a computationally intensive process that consists of two phases: during the training (TR) step, a set of parameters is found [8]; during the Model Selection (MS) phase, a set of additional variables (hyperparameters) is tuned to find the SVM characterized by optimal performance in classifying previously unseen data [1]. In case of linear SVM classifiers, for example, the set of hyperparameters is $\mathcal{C} = \{C\}$, where $C$ weights a regularization term [10].

With the growth of Online Learning (OL) applications (involving large data streams with varying distribution) [**?** ], the naïve SVM is incapable of handling such problems without any modifications: in fact, in the OL framework model update is necessary every time a new sample is gathered [2, 3], but a complete MS-TR run from scratch becomes computationally prohibitive. While incremental TR algorithms have been developed in order to incorporate additional training samples [3, 7], the extension to the OL framework of MS approaches is not as much straightforward, despite representing a desirable feature [5, 7].

In this paper, we thus focus on the MS step of SVM learning in the OL framework, by proposing an efficient heuristic approach for updating the set of hyperparameters $\mathcal{C}$ when new samples are collected.

## 2 MS in the OL framework

Let us consider a continuos online stream of data $D_t = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_t, y_t), \ldots\}$, $\boldsymbol{x}_i \in \Re^d$, $y_i = \pm 1$ where the relation between $\boldsymbol{x}$ and $y$ is encapsulated by an unknown distribution $\mathcal{P}_t$ that can change over time. The goal is to exploit newly gathered data, obtained from the stream, in order to learn, or improve, a function $f_t$ that approximates $\mathcal{P}_t$.

Typically, in the OL framework, MS is performed only once in a batch mode by exploiting the first $t$ data collected: by using a conventional MS approach, such as the K-Fold Cross Validation (KCV), the SVM hyperparameters are fixed to the optimal value at the $t$-th step $\mathcal{C}_t^*$ and a first model is trained $(f_t)$. When $\Delta$ new samples are gathered at step $t + \Delta$, the learning set is updated $D_{t+\Delta} = D_t \setminus \{(\boldsymbol{x}_1, y_1), ..., (\boldsymbol{x}_\Delta, y_\Delta)\} \cup \{(\boldsymbol{x}_{t+1}, y_{t+1}), ..., (\boldsymbol{x}_{t+\Delta}, y_{t+\Delta})\}$ and $f_{t+\Delta}$ is consequently modified, e.g. accordingly to the method proposed in [3]. As $\mathcal{C}_t^*$ is kept constant in this process, we define this OL approach as NO-MS.

Ideally, instead, we should both be able to update the model and the set of hyperparameters thank to the new collected samples. A possible, but computationally expensive, approach consists in performing a full MS-TR procedure at every step, i.e. a complete learning from scratch every time new patterns are collected: we define this approach as Complete-MS (C-MS).

In this work, we propose a new heuristic approach, where MS is not completely neglected in OL, as in NO-MS, but a whole re-learning is avoided, contrarily to C-MS. Analogously to NO-MS, we start by identifying the best hyperparameters set $\mathcal{C}_t^*$ and model $f_t$ at the $t$-th step with a KCV procedure; differently from NO-MS, we also have to keep track of the $k$ models $f_t^{(1)}, \ldots, f_t^{(k)}$ trained while applying KCV. When $\Delta$ new samples are acquired, we want both to modify the hyperparameters set and the model: it is reasonable to assume that, if $\Delta$ is not large (i.e. it is not comparable to $t$), the hyperparameters will not vary too much from the previous best value. Thus, we can define a neighborhood

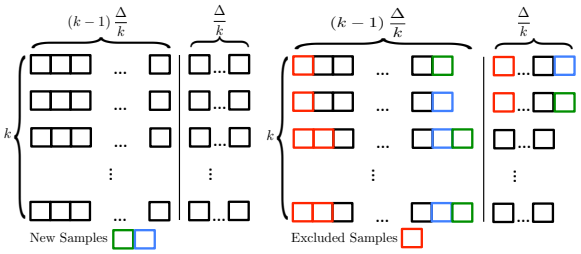



**Fig. 1:** Online k–Fold Cross Validation.

set $\mathcal{C}_t^\epsilon$, centered around $\mathcal{C}_t^*$: for example, if linear kernels are concerned, we can define a neighborhood set $C \in \left[\frac{C_t^*}{\epsilon}, \epsilon C_t^*\right]$, where $\epsilon > 1$. We can update the KCV samples set (refer to Fig. 1) and, accordingly, the KCV models $f_{t+\Delta}^{(1,\dots,k)}$ by exploiting the approach proposed in [4] for every value of the hyperparameters included in the neighborhood set: through a conventional MS approach based on KCV, we can thus identify the best hyperparameters configuration $\mathcal{C}_{t+\Delta}^* \in \mathcal{C}_t^\epsilon$. Finally, $f_{t+\Delta}$ is trained accordingly to $\mathcal{C}_{t+\Delta}^*$ and the $\Delta$ new samples collected by exploiting the procedure described in [4]. We define this approach as OL-MS.

The supplementary computational burden with respect to NO-MS is limited: in fact, we have to update $k+1$ models instead of 1 and we have to perform a KCV MS at each updating step, but limited to a restricted neighborhood set; however, the OL-MS approach copes with (reasonably slow) modifications of $\mathcal{P}_t$, allowing to properly re-tune the models and the hyperparameters configuration and, thus, representing an effective trade-off between the computationally unfeasible C-MS and the unadaptive NO-MS procedures.

## 3 Preliminary Results and Discussion

We compare the three approaches, introduced above, by using the Spam dataset [6], consisting of 9324 samples and $d = 39917$ features. Since $d \gg n$, a linear classifier is sufficient to separate the samples (i.e. $\mathcal{C} = \{C\}$). We use an initial batch of $t = 1000$ samples for the first learning, and we perform OL by adding $\Delta = 50$ samples per iteration. We look for the hyperparameter $C$ in the range $\left[10^{-5}, 10^2\right]$ by using 50 points equally spaced in logarithmic scale, while we fix $k = 4$ for the KCV procedure. We also set $\epsilon = 10$ in OL–MS.

For the three approaches presented, Fig. 2 shows the error rates on the test set, consisting of the next $10\Delta$ samples not yet exploited for OL, while Tab. 3 presents the learning times on a conventional PC. Results clearly show that the proposed OL-MS procedure represents a good trade-off between accuracy and training time: while the former is comparable to the one of C-MS, the latter are more than one order of magnitude smaller than the time needed by C-MS and acceptable in several OL applications. These results are encouraging, nevertheless the OL-MS approach requires that extensive simulations are performed to assess its performance and to study the effect of variations of parameters (e.g. $\epsilon$) on the quality of the results (both in terms of error rate and training time).

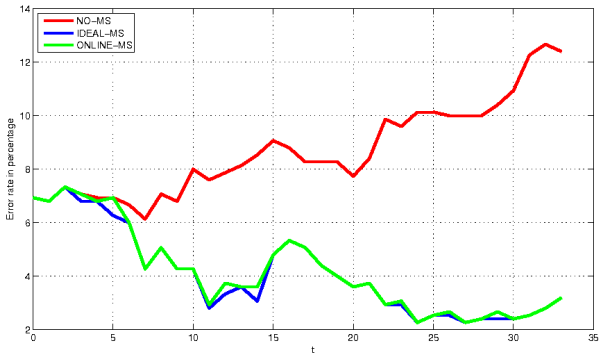



**Fig. 2:** Comparison of error rates on the test set as new patterns are considered.

| Method | Training Time (seconds) |
|---|---|
| NO–MS | $0.60 \pm 0.25$ |
| C–MS | $43.92 \pm 1.31$ |
| OL–MS | $2.57 \pm 0.35$ |

**Tab. 1:** Training times for the different OL approaches.

## References

[1] D. Anguita, A. Ghio, L. Oneto, and S. Ridella. In-sample and out-of-sample model selection and error estimation for support vector machines. *IEEE Transactions on Neural Networks and Learning Systems*, 23(9):1390–1406, 2012.

[2] A. Blum. On-line algorithms in machine learning. *Online algorithms*, pages 306–325, 1998.

[3] G. Cauwenberghs and T. Poggio. Incremental and decremental support vector machine learning. *Advances in neural information processing systems*, pages 409–415, 2001.

[4] C. P. Diehl and G. Cauwenberghs. Svm incremental learning, adaptation and optimization. In *IEEE International Joint Conference on Neural Networks*, pages 2685–2690, 2003.

[5] T. Diethe and M. Girolami. Online learning with (multiple) kernels: A review. *Neural computation*, 25(3):567–625, 2013.

[6] I. Katakis, G. Tsoumakas, and I. Vlahavas. Tracking recurring contexts using ensemble classifiers: an application to email filtering. *Knowledge and Information Systems*, 22(3):371–391, 2010.

[7] P. Laskov, C. Gehl, S. Krüger, and K.R. Müller. Incremental support vector learning: Analysis, implementation and applications. *The Journal of Machine Learning Research*, 7:1909–1936, 2006.

[8] J. Shawe-Taylor and S. Sun. A review of optimization methodologies in support vector machines. *Neurocomputing*, 74(17):3609–3618, 2011.

[9] J.A.K. Suykens and J. Vandewalle. Least squares support vector machine classifiers. *Neural processing letters*, 9(3):293–300, 1999.

[10] V. Vapnik. *Statistical Learning Theory*. Wiley New York, 1998.

# Lasso and Adaptive Lasso with Convex Loss Functions

Wojciech Rejchel
Nicolaus Copernicus University
Chopina 12/18, 87-100 Toruń, Poland
wrejchel@gmail.com

**Abstract**: Variable selection is a fundamental challenge in statistical learning if one works with data sets containing huge amount of predictors. In the paper we consider procedures popular in model selection: Lasso and adaptive Lasso. Our goal is to investigate properties of estimators based on minimization of Lasso-type penalized empirical risk with a convex loss function, in particular nondifferentiable. We obtain theorems concerning rate of convergence in estimation, consistency in model selection or oracle properties for Lasso estimators if the number of predictors is fixed, i.e. it does not depend on the sample size. Moreover, we study properties of Lasso and adaptive Lasso estimators on simulated and real data sets.

**Keywords**: Lasso, Adaptive Lasso, Model Selection, Oracle Property, Convex Loss Function

Variable selection is a fundamental challenge in statistical learning if one works with data sets containing huge amount of predictors. Such problems often occur in computational biology, medicine or banking. Finding significant (relevant) variables helps to better understand the problem and improves statistical inference. One of methods solving this problem is Lasso [7]. The main characteristic of this procedure is an ability to select significant variables and estimate unknown parameters simultaneously if the penalty term is chosen in an appropriate way. Our goal is to investigate properties of estimators based on minimization of such penalized empirical risk with a convex loss function.

In the paper we consider a model with $X$ being a $d$-dimensional random vector and $Y$ a random variable. For convenience we assume that $X \in \mathbb{R}^d$ and $Y \in \mathbb{R}$. We treat $X$ as a vector of predictors and $Y$ as a response variable. The dependence between $Y$ and $X$ is contained in a parameter $\theta^* \in \mathbb{R}^d$. We think of the $j$-th variable being relevant in the considered model if $\theta_j^* \neq 0$ and irrelevant otherwise. Let relevant predictors be determined by a set $\mathcal{A} = \{1, \ldots, d_0\}$ for some $0 < d_0 < d$, so our model depends on a subset of all predictors. Denote $\theta_{\mathcal{A}}^* = \{\theta_1^*, \ldots, \theta_{d_0}^*\}$.

Let $f(\theta, y, x)$ be a real loss function. We assume that $f$ is convex with respect to $\theta$ for fixed $z = (y, x)$ and measurable with respect to $z$ for fixed $\theta$. Consider the convex function $Q(\theta) = \mathbb{E}f(\theta, Z)$ and its minimizer $\theta^*$. If we do not know the distribution of $Z$, then we cannot calculate $\theta^*$ directly. However, if we have a sample $Z_1, \ldots, Z_n$ of independent copies of $Z$, then we can minimize $Q_n(\theta) = \frac{1}{n}\sum_{i=1}^n f(\theta, Z_i)$ instead of $Q(\theta)$. This standard approach is often modified by adding some penalty to the empirical risk $Q_n$. Considering Lasso we are to minimize a convex in $\theta$ function $\Gamma_n(\theta) = Q_n(\theta) + \frac{\lambda_n}{n}|\theta|$, where $|\cdot|$ is the $l_1$-norm of the vector $\theta$, i.e. $|\theta| = \sum_{j=1}^d |\theta_j|$ and $\lambda_n > 0$ is a number dependent on $n$ that is chosen by a researcher. It is a balance between minimizing the empirical risk and the penalty. We denote the minimizer of $\Gamma_n(\theta)$ by $\hat{\theta}$. The form of the penalty is crucial, because its singularity at the origin implies that some coordinates of the minimizer $\hat{\theta}$ are exactly equal to zero if $\lambda_n$ is sufficiently large. Thus, by minimizing the function $\Gamma_n(\theta)$ we estimate unknown parameters and select significant predictors simultaneously.

Lasso is very popular in model selection and estimation. Its theoretical properties for linear models with the quadratic loss function or generalized linear models were widely studied [1, 3, 4, 9, 10]. However, there is a comprehensive literature where one uses Lasso estimators with different loss functions, for instance the absolute value or the "hinge" loss well-known in machine learning. There is a natural question if using more appropriate in investigated model non-quadratic loss function we can expect that the estimator behaves in model selection similarly to the relatively well-studied quadratic loss case. In our paper we describe properties of estimators based on minimization of Lasso-type penalized empirical risk with a convex loss function, in particular this loss function can be nondifferentiable. Our results generalize theorems from the abovementioned papers if $d$ is fixed, i.e. it does not depend on $n$. While working with the convex setting we need some standard regularity assumptions:

(a) $\theta^*$ is unique,

(b) $Q$ is twice differentiable at $\theta^*$ and $H = \nabla^2 Q(\theta^*)$ is positive definite,

(c) $\mathbb{E}|\partial f(\theta, Z)|^2 < \infty$ for each $\theta$ in some neighbourhood of $\theta^*$. We do not require that the subgradient $\partial f(\theta, z)$ of a convex function $f(\theta, z)$ is unique, we only require that it is a measurable selection of the subgradient.

We prove (Theorem 0.1) that considered estimator cannot be an "oracle", because if the rate in estimation is optimal then the consistency in model selection is not possible. Denote $\mathcal{A}_n = \{j \in \{1, \dots, d\} : \hat{\theta}_j \neq 0\}$.

**Theorem 0.1.** *If* $\lambda_n/\sqrt{n} \to \lambda_0 \geq 0$, *then*
*(a)* $\sqrt{n}\left(\hat{\theta} - \theta^*\right) \to_d \arg\min_\theta V(\theta)$, *where*
$V(\theta) = \frac{1}{2}\theta^T H\theta + W^T\theta + \lambda_0 \sum_{j\in\mathcal{A}} \theta_j \mathrm{sign}(\theta_j^*) + \lambda_0 \sum_{j\notin\mathcal{A}} |\theta_j|$
*and* $W \sim N(0, D)$ *with* $D = \mathbf{Var}\partial f(\theta^*, Z)$,
*(b)* $\limsup_n \mathbb{P}(\mathcal{A}_n = \mathcal{A}) \leq K < 1$, *where* $K$ *is a constant.*

Moreover, we show that even if one agrees on the worse rate in estimation, then consistency in model selection is possible but not guaranteed. We obtain necessary and sufficient conditions (Theorem 0.2) for consistency in model selection in this case.

**Theorem 0.2.** *Suppose that* $\frac{\lambda_n}{n} \searrow 0$, *and* $\frac{\lambda_n}{\sqrt{n}} \to \infty$.
*(a) If Lasso estimator is consistent in model selection, then the inequality*

$$\left|H_2^T H_1^{-1} \mathrm{sign}(\theta_{\mathcal{A}}^*)\right| \leq 1,$$

*holds componentwise and matrices* $H_1$ *and* $H_2$ *are taken from the matrix*

$$H = \begin{pmatrix} \overbrace{H_1}^{d_0\times d_0} & \overbrace{H_2}^{d_0\times(d-d_0)} \\ H_2^T & H_3 \end{pmatrix}.$$

*(b) If the inequality*

$$\left|H_2^T H_1^{-1} \mathrm{sign}(\theta_{\mathcal{A}}^*)\right| < 1$$

*holds componentwise, then Lasso estimator is consistent in model selection.*

Furthermore, we describe an improvement of Lasso that was proposed in [10] and called "adaptive Lasso". It relies on adding different weights to different predictors in the penalty. Namely, we are to minimize the function $\Gamma_n^a(\theta) = Q_n(\theta) + \frac{\lambda_n}{n}\sum_{j=1}^d \frac{|\theta_j|}{|\hat{\beta}_j|}$, where $\hat{\beta}$ is an arbitrary estimator of $\theta^*$ such that $\sqrt{n}\left(\hat{\beta} - \theta^*\right) = O_P(1)$. We prove that the oracle property holds in the convex setting that extends previous theorems [8, 10].

**Theorem 0.3.** *If* $\lambda_n \to \infty$ *and* $\frac{\lambda_n}{\sqrt{n}} \to 0$, *then the adaptive lasso estimator* $\hat{\theta}^a = \arg\min_\theta \Gamma_n^a(\theta)$ *is an oracle:*
*(a)* $\lim_{n\to\infty} \mathbb{P}\left(\mathcal{A}_n^a = \mathcal{A}\right) = 1$,
*(b)* $\sqrt{n}\left(\hat{\theta}_{\mathcal{A}}^a - \theta_{\mathcal{A}}^*\right) \to_d N\left(0, H_1^{-1} D_1 H_1^{-1}\right)$ *and the matrix* $D_1$ *is a* $(d_0 \times d_0)$ *upper-left submatrix of* $D$.

Notice that convexity of the loss function and methods from the convex empirical process theory [2, 5, 6] play crucial roles in our argumentation. Finally, we discuss regularity assumptions in the non-trivial case that the loss function is non-differentiable, for instance $f(\theta, z) = |y - \theta^T x|$. If we want the risk $Q(\theta) = \mathbb{E}|Y - \theta^T X|$ to be twice differentiable at $\theta^*$, we need some regularity assumptions on the probability distribution of $(Y, X)$. Consider the standard linear model $Y = (\theta^*)^T X + \varepsilon$, where $\mathbb{E}|X|^2 < \infty$, besides $\varepsilon$ and $X$ are independent. Then we only need that $\varepsilon$ has a density $l(\cdot)$ continuous and positive in a neighbourhood of zero. In this case one can calculate that $H = 2\,l(0)\,\mathbb{E}XX^T$ and $D = \mathbb{E}XX^T$ (see [6]).

## References


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

[2] C. J. Geyer. On the asymptotics of convex stochastic optimization. *Unpublished manuscript*, 1996.

[3] K. Knight and W. Fu. Asymptotics for lasso-type estimators. *Annals of Statistics*, 28:1356–1378, 2000.

[4] N. Meinshausen and P. Buhlmann. High-dimensional graphs and variable selection with the lasso. *Annals of Statistics*, 34:1436–1462, 2006.

[5] W. Niemiro. Asymptotics for M-estimators defined by convex minimization. *Annals of Statistics*, 20:1514–1533, 1992.

[6] D. Pollard. Asymptotics for least absolute deviation regression estimators. *Econometric Theory*, 7:186–199, 1991.

[7] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1996.

[8] H. Wang, G. Li, and G. Jiang. Robust regression shrinkage and consistent variable selection through the lad-lasso. *Journal of Business & Economic Statistics*, 25:347–355, 2007.

[9] P. Zhao and B. Yu. On model selection consistency of lasso. *Journal of Machine Learning Research*, 7:2541–2563, 2006.

[10] H. Zou. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101:1418–1429, 2006.

# Conditional Gaussian Graphical Models for Multi-output Regression of Neuroimaging Data

Andre F Marquand
King's College London
andre.marquand@kcl.ac.uk

Maria Joao Rosa
University College London
m.rosa@ucl.ac.uk

Orla Doyle
King's College London
orla.doyle@kcl.ac.uk

**Abstract**: Pattern recognition has shown considerable promise for automated diagnosis and for predicting outcome in clinical neuroimaging studies. Many of these decision problems can most naturally be framed as multi-output learning problems but nearly all studies to date have adopted sub-optimal analytical approaches, modeling different outputs independently. In this work, we apply a dual elastic net regularisation to the precision matrix of a conditional Gaussian graphical model to provide true multi-output learning for clincial neuroimaging data. This method improves the accuracy over modeling the outputs independently and can quantify the relationships between outputs, which is useful to help understand disease pathology.

**Keywords**: Multi-output learning, multi-task learning, conditional Gaussian graphical models, neuroimaging, MRI, multiple system atrophy

## 1 Introduction

Pattern recognition has shown to be highly promising for clinical neuroimaging research both for computer-assisted diagnosis and for predicting future disease outcome. To date, nearly all applications have focussed on binary classification, with only a few applications of (single output) regression. These approaches are over-simplifications for most clinical problems, where it may be necessary to predict multiple outcomes to adequately describe a disease state (e.g. to measure different types of symptoms). Multi-output learning is thus a natural candidate for this type of problem, but has received little attention from the neuroimaging community. Here, we demonstrate its suitability for neuroimaging. We extend a recently proposed approach based on conditional Gaussian graphical models (CGGMs) where we enforce sparsity on the precision matrix using a dual elastic net penalty, derive patterns of predictive weights that are coupled between outputs and automatically estimate correlations between the outputs. We apply this approach to a challenging and clinically relevant problem: predicting clinical scales from neuroimaging data derived from patients with a fatal neurological disorder.

## 2 Methods

We begin with a dataset $\{\mathbf{X}, \mathbf{Y}\}$, where $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_n]^T$ is an $n \times d$ matrix containing $d$-dimensional data vectors. Each data vector is associated with an $m$-dimensional vector of real-valued outputs, which are collected in the $n \times m$ matrix $\mathbf{Y} = [\mathbf{y}_1, ..., \mathbf{y}_n]^T$. The goal is to simultaneously predict the outputs while properly accounting for the dependencies between them. Conventionally, this is achieved by generalising single output regression, using a different weight vector to predict each output, i.e. $\mathbf{Y} = \mathbf{XB} + \mathbf{E}$, where $\mathbf{B}$ is a $d \times m$ weight matrix and $\mathbf{E}$ is an $n \times m$ residual matrix. Most existing approaches to estimate $\mathbf{B}$ have limitations in that they either: (i) account for dependencies between the weight vectors or between the targets, but not both or (ii) require the dependency structure between outputs to be specified *a priori*, which requires knowledge that is unlikely to be available for many applications. In this work, we generalise a recently proposed approach that overcomes these limitations [4]. Here, the problem is recast in a CGGM framework which assumes a joint Gaussian distribution over the inputs and targets, i.e. $[\mathbf{x}_i^T, \mathbf{y}_i^T]^T \sim \mathcal{N}\left(\mathbf{0}, \begin{pmatrix} \mathbf{\Sigma_{xx}} & \mathbf{\Sigma_{xy}} \\ \mathbf{\Sigma_{xy}}^T & \mathbf{\Sigma_{yy}} \end{pmatrix}\right)$. The aim is then to estimate the entries of the precision matrix: $\mathbf{\Theta} = \mathbf{\Sigma}^{-1} = \begin{pmatrix} \mathbf{\Theta_{xx}} & \mathbf{\Theta_{xy}} \\ \mathbf{\Theta_{xy}}^T & \mathbf{\Theta_{yy}} \end{pmatrix}$, which is a natural parameterisation because of the well-known fact that zeros in the precision matrix encode the conditional independence structure between variables. Finally, predictions can be made by $p(\mathbf{y}_*|\mathbf{x}_*) \sim N(-\mathbf{\Theta_{yy}}^{-1}\mathbf{\Theta_{xy}}^T\mathbf{x}_*, \mathbf{\Theta_{yy}}^{-1})$. It turns out that this representation has a correspondence to the conventional formulation, with $\mathbf{B} = -\mathbf{\Theta_{yy}}^{-1}\mathbf{\Theta_{xy}}^T$ [4]. Model parameters can be estimated by considering the conditional log-likelihood:

$$L(\mathbf{X}, \mathbf{Y}|\mathbf{\Theta_{xy}}, \mathbf{\Theta_{yy}}) = -0.5 \left[-n \log\det(\mathbf{\Theta_{yy}}) + \right.$$
$$\left. \mathrm{tr}\left[(\mathbf{Y} + \mathbf{X}\mathbf{\Theta_{xy}}\mathbf{\Theta_{yy}}^{-1})\mathbf{\Theta_{yy}}(\mathbf{Y} + \mathbf{X}\mathbf{\Theta_{xy}}\mathbf{\Theta_{yy}}^{-1})^T\right]\right]$$

Maximum likelihood estimation of this model is equivalent to estimating each output independently, but applying regularisation allows the weights for each output

to borrow strength from one another. We apply two elastic net penalties, which forces some entries of $\mathbf{\Theta_{xy}}$ and $\mathbf{\Theta_{yy}}$ to zero, while retaining correlated features. This last property is important because neuroimaging data is characterised by substantial spatial and temporal correlation. Therefore, we aim to maximise: $L(\mathbf{X}, \mathbf{Y}|\mathbf{\Theta_{xy}}, \mathbf{\Theta_{yy}}) - \lambda_1[\alpha_1|\mathbf{\Theta_{xy}}|_1 + (1-\alpha_1)\|\mathbf{\Theta_{xy}}\|_F^2] - \lambda_2[\alpha_2|\mathbf{\Theta_{yy}}|_1 + (1-\alpha_2)\|\mathbf{\Theta_{yy}}\|_F^2]$, where $|.|_1$ is a matrix L1 norm, $\|.\|_F$ is the Frobenius norm and $\alpha_j$ and $\lambda_j$ are regularisation hyperparameters. This is a convex optimisation problem (see [4]), which we solve using a projected scaled subgradient method. Note that it is not necessary to explicitly estimate $\mathbf{\Theta_{xx}}$, which is advantageous because neuroimaging data are often very high dimensional.

We apply this method to predicting clinical scores from neuroimaging data from patients with a motor neurodegenerative disorder (see [1] for details). Here, we use the structural neuroimaging data from 19 patients with multiple system atrophy. We aim to predict a set of eight standard clinical scales: Universal Parkinson's disease rating scale 1 (UPDRS1 - mentation and mood), UPDRS2 (activities of daily living), UPDRS3 (motor function), PIGD (postural stability), the short motor disability scale (SMDS), Hoehn and Yahr (HY - measuring disease staging) and the Schwab and England activities of daily living scale (SE-ADL). This is a highly challenging problem because the scales are calibrated across different ranges. We use nested cross-validation with a grid search to optimise the hyperparameters. While CGGM is computationally tractable for whole-brain prediction, a full grid search becomes quite computationally demanding. Therefore, we reduce the dimensionality using agglomerative clustering [2]. We construct features from a set of "scalar momentum" image features [3], with the number of clusters fixed to 200 (100 grey- and 100 white matter).

## 3 Results

The error obtained by the CGGM on all symptom scales is reported in Table 1 along with a baseline model that predicts each variable independently (ridge regression). The CGGM produced equivalent or better performance for seven of the eight scales. A second important outcome is an estimate of the partial correlation between variables ($\rho$), which can be derived from $\mathbf{\Theta}_{yy}$ (Fig 1) and is useful to quantify the relationships between output variables. A third outcome is a visualisation of the predictive weights in the original voxel space, which is particularly crucial for neuroimaging, but a lack of space precludes its presentation here. Note that the optimal parameters from the grid search favoured a largely non-sparse model for both $\mathbf{\Theta_{yy}}$ and $\mathbf{\Theta_{xy}}$. Therefore a ridge penalty would probably perform equivalently for this data, although this would have been difficult to determine in advance.

| Clinical Scale | Ridge Regression | CGGM |
|---|---|---|
| UPDRS 1 | 1.28 | **1.14** |
| UPDRS 2 | 1.05 | **0.99** |
| UPDRS 3 | 0.61 | **0.51** |
| PIGD | 1.01 | **0.92** |
| Stability | **0.66** | 0.74 |
| SMDS | 0.62 | **0.59** |
| Hoehn-Yahr | **0.72** | **0.72** |
| Schwab-England | 0.73 | **0.65** |

**Tab. 1:** Standardized mean squared error for the CGGM and a baseline model that predicts each target separately

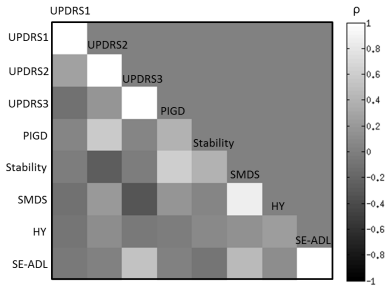



**Fig. 1:** Partial correlation ($\rho$) between clinical variables

## 4 Conclusions

We have demonstrated a flexible multi-output learning approach based on CGGMs for neuroimaging data. This approach produced highly promising performance on the clinical dataset investigated and provides measures that are useful to help understand the relationships between different outputs and how well they can be predicted from the neuroimaging data.

**Acknowledgments**
AFM was supported by the KCL Medical Engineering Centre, funded by the Wellcome Trust and EPSRC.

## References

[1] M. Filippone, A. Marquand, C. Blain, Steven Williams, J. Mourao-Miranda, Mark Girolami, Probabilistic prediction of neurological disorders with a statistical assessment of neuroimaging data modalities *Annals of Applied Statistics*, 6: 1883–1905, 2012.

[2] V. Michel, A. Gramfort, G. Varoquaux, E. Eger, C. Keribin, B. Thirion. A supervised clustering approach for fMRI-based inference of brain states. *Pattern Recognition*, 45(6): 2041–2049, 2012.

[3] N. Singh, P. Fletcher, J. Preston, L. Ha, R. King, J. Marron, M. Wiener, S. Joshi. Multivariate statistical analysis of deformation momenta relating anatomical shape to neuropsychological measures. *Lecture Notes in Computer Science*, 6363:529–537, 2010

[4] K. Sohn and S. Kim. Joint estimation of structured sparsity and output structure in multiple-output regression via inverse-covariance regularization *Journal of Machine Learning Research - Proceedings Track*, 22:1081–1089, 2012.

# High-dimensional convex optimization via optimal affine subgradient algorithms

Masoud Ahookhosh
Faculty of Mathematics
University of Vienna
masoud.ahookhosh@univie.ac.at

Arnold Neumaier
Faculty of Mathematics
University of Vienna
Arnold.Neumaier@univie.ac.at

**Abstract**: This study is concerned with some algorithms for solving high-dimensional convex optimization problems appearing in applied sciences like signal and image processing, machine learning and statistics. We improve an optimal first-order approach for a class of objective functions including costly affine terms by employing a special multidimensional subspace search. We report some numerical results for some imaging problems including nonsmooth regularization terms.

## 1 Introduction and Basic Idea

Lots of applications in signal and image processing, machine learning, statistics, geophysics, etc include affine terms in their structure that are the most costly part of function evaluations. Therefore, we consider the following unconstrained convex optimization problem

$$\text{minimize } f(x) := \sum_{i=1}^{n_1} f_i(x, A_i(x)), \tag{1}$$

where $f_i(x, v_i)$ $(i = 1, \cdots, n_1)$ are convex functions defined for $x \in \mathbb{R}^n$ and $v_i = A_i(x) \in \mathbb{R}^{m_i}$ and the $A_i$ are linear operators from $\mathbb{R}^n$ to $\mathbb{R}^{m_i}$. The aim is to derive an approximating solution $x$ just using function values $f(x)$ and subgradients $g(x)$. Consider the following generic first-order descent algorithm:

**Generic first-order descent algorithm**

Input: $x_0 \in \mathbb{R}^n$; $\epsilon > 0$;
Output: $x_b$;
**Begin**
1 choose $x_b$;
2 calculate $f_{x_b} \leftarrow f(x_b)$; $g_{x_b} \leftarrow g(x_b)$;
3 **Repeat**
4 generate $x_b'$ by an extended algorithm;
5 generate $\bar{x}_b$ by a heuristic algorithm;
6 $x_b \leftarrow \operatorname{argmin}_{\{x_b, x_b', \bar{x}_b\}} f(x)$;
7 **Until** a stopping criterion holds;
**End**

On the basis of high-dimensional data used in applications, the most costly part of function and subgradient evaluations relates to applying the linear operator and its transpose. Then we should decrease the number of linear operator calculations as much as possible. To find an appropriate $\bar{x}_b$ in line 5 of this algorithm, we use in each step of this algorithm a multidimensional subspace search as inner algorithm. The main idea is a generalization of line search techniques by saving $M \ll n$ columns $u$ and the previously computed vectors $v_i = A_i(u)$ in the matrices $U$ and $V_i$ $(i = 1, \cdots, n_1)$, find $t^* \approx \operatorname{argmin}_{t \in \mathbb{R}^M} \tilde{f}(t)$ where

$$\tilde{f}(t) := f(x_b + Ut) = \sum_{i=1}^{n_1} f_i(x + Ut, v_i + V_i t) \tag{2}$$

can be calculated without further calls to the $A_i$.

**Multidimensional subspace search procedure**

Input: $x_b \in \mathbb{R}^n$; $U$; $v_i \in \mathbb{R}^{m_i}$; $V_i (i = 1, \cdots, n_1)$;
**Begin**
approximately solve the $M$-dimensional problem

$$t^* \approx \operatorname*{argmin}_{t \in \mathbb{R}^M} \tilde{f}(t); \tag{3}$$

$\bar{x}_b = x_b + Ut$;
**End**

## 2 Specific version of the algorithm

The general idea is made specific by choosing a particular descent algorithm. We use the OSGA algorithm, [3], which monotonically reduces a bound on the error $f(x_b)$ of the function value of the best known point $x_b$. The modified version of OSGA including affine scaling and the multidimensional subspace search, as defined above, is called ASGA which is suitable to deal with high-dimensional convex optimization appearing in applications. In ASGA, the subproblem (3) is also solved by OSGA.

The OSGA algorithm generates and updates linear relaxations

$$f(z) \geq \gamma + h^T z \quad \forall z \in \mathbb{R}^n,$$

where $\gamma \in \mathbb{R}$, $h \in \mathbb{R}^n$. In line 3 of the algorithm we set $x_b' = x_b + \alpha(u - x_b)$, where $u = U(\gamma, h) \in C$ solves a minimization problem of the form

$$E(\gamma, h) := -\inf_{x \in \mathbb{R}^n} \frac{\gamma + h^T z}{Q_0 + \frac{1}{2}\|z - z_0\|^2}. \tag{4}$$

For details see [3]. It is proved in [3] that OSGA achieves the optimal complexity bounds $O(1/\sqrt{\epsilon})$ for the optimization of smooth convex functions and $O(1/\epsilon^2)$ for nonsmooth convex functions [2], no matter which heuristic choice $\bar{x}_b$ is made, so ASGA is an optimal complexity algorithm.

## 3 Numerical Results

This section reports some numerical results to show the efficiency of the proposed algorithms for solving practical problems arising in applications. On the basis of the fact that OSGA and ASGA just need function and subgradient evaluations, they can be employed in solving wide range of problems including regularization terms. Examples include *lasso, basis pursuit denoising,* $l_1$ *and* $l_2$ *decoding, isotropic and anisotropic total variation, group regularizations, elastic net, nuclear norm, linear support vector machine, kernel-based models.*

Here, we consider the $l_2^2 - ITV$ regularization problem defined by

$$f(x) = \frac{1}{2}\|A(x) - b\|_2^2 + \lambda\|x\|_{ITV},$$

where $\|.\|_{ITV}$ denotes the isotropic $TV$ norm and $A$ is a linear operator chosen as in the TwIST package for reconstruction of the $512\times512$ blurry-noisy Lena image.

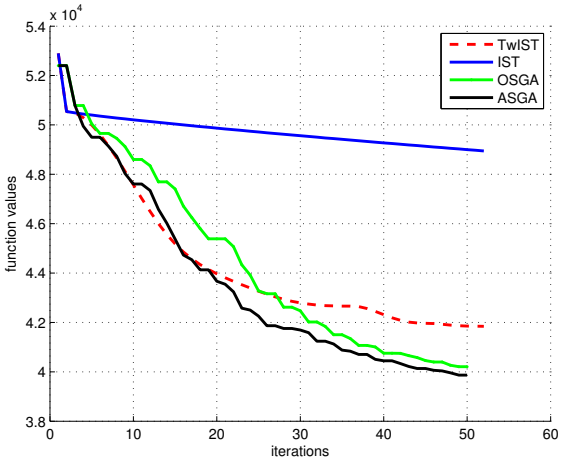



**Fig. 1:** A comparison of function values for IST, TwIST, OSGA and ASGA

If we count the images of Fig. 2 row by row, the first image is the original image, and the second image shows a blurry-noisy image constructed by adding noise and uniform $9 \times 9$ blur with BSNR $= 40$ dB. The rest of the images are restored by the minimization problem (3) where $x_0$ is given by a Wiener filter for all considered algorithms. The third image is produced by the IST algorithm, while the fourth image is recovered by TwIST [4]. Also the fifth image restored by OSGA and the sixth images was reconstructed by ASGA. It is clear that the final function value of the proposed algorithms




**Fig. 2:** Isotropic TV-based image reconstruction from 40% missing sample with the considered algorithms: IST, TwIST, OSGA and ASGA

is less than those of IST and TwIST and the restored image visually looks good. This shows that the proposed algorithms can effectively reconstruct blurry and noisy images at a reasonable cost.

## References

[1] M. Ahookhosh, A. Neumaier, An affine subgradient algorithm for large-scale convex optimization. *Manuscript*, University of Vienna, 2013.

[2] Y. Nesterov, Introductory lectures on convex optimization: A basic course, Kluwer, Dordrecht 2004.

[3] A. Neumaier, Convex optimization with optimal complexity. *Manuscript*, University of Vienna, 2013.

[4] TwIST website: http://www.lx.it.pt/ bioucas/TwIST/TwIST.htm

# Joint Estimation of Modular Gaussian Graphical Models

José Sánchez
Mathematical Sciences, Chalmers University of Technology and University of Gothenburg,
SE-412 96 Gothenburg, Sweden
sanchezj@chalmers.se

Rebecka Jörnsten
Mathematical Sciences, Chalmers University of Technology and University of Gothenburg,
SE-412 96 Göteborg, Sweden
jornsten@chalmers.se

**Abstract**: We propose a method to estimate several Gaussian graphical models that share a common structure with modular topology. To encourage modularity we define a novel adaptive fused penalty. We also propose a generalization of the fused penalty to a more general one defined by graphs. We use this penalty to correct for unequal sample sizes in the data or integrate ordered variables. We optimize the penalized log-likelihood using ADMM and show by simulation that our method performs better than competitors. We apply our method in the study of regulators in glioblastoma, breast and ovarian cancer.

**Keywords**: Gaussian graphical models, gene networks, lasso, fused lasso, elastic net, precision matrix estimation, data integration.

## 1 Introduction

Modeling of transcription networks is a popular approach to cancer research at molecular level. It has been used to integrate several types of genomic cancer data, identify genes with altered copy number as disease drivers, construction of features for prediction of patient survival and identification of potential therapeutic targets as hubs [3].

Assume that transcription information (mRNA) of class $k$ (such as a cancer type), $k = 1, 2, \ldots, K$, can be modeled as a realization of a multivariate normal distribution with mean $\mu^k$ and covariance matrix $\Sigma^k$. In this case, the problem of estimating the network is equivalent to estimate the precision matrix $\Omega^k = \left(\Sigma^k\right)^{-1}$.

To jointly estimate these networks, we optimize a penalized version of the log-likelihood function using ADMM [1]. We include an *elastic net* penalty in order to obtain sparse networks for ease of interpretation. Since it is biologically reasonable to assume that some of the classes will share regulators, we also include a *fused* penalty, to achieve equality across classes. Finally, we believe that this equality constraint should be smooth at least locally, thus we defining a *module* or a *modular* network structure. To encourage modularity we define an adaptive fused penalty through an adaptivity factor. This adaptivity factor is computed from an initial zero-consistent [5] solution, that encourages fusion for neighborhoods of links.

## 2 Methods and Results

Consider $K$ data sets $X^1$, $X^2, \ldots, X^K$ with $K \geq 2$ corresponding to $K$ classes. Data set $X^k$ consists of $n_k$ observations and $p$ variables, which are common to all $K$ classes. We assume the observations within each data set to be i.i.d. $N(0, \Sigma^k)$. Let $\Theta^k = \left(\Sigma^k\right)^{-1}$ and $S^k$ be the empirical covariance matrix of the $k$ class. We propose, similarly to [2], to optimize the penalized log-likelihood function

$$\begin{aligned} l\left(\{\Theta\}\right) = &\sum_{k=1}^{K} n_k \left[\ln\left(\det\left(\Theta^k\right)\right) - \operatorname{tr}\left(S^k\Theta^k\right)\right] \\ &- \lambda_1 \sum_{k=1}^{K}\sum_{i\neq j}\left[\alpha\left|\theta_{ij}^k\right| + (1-\alpha)\left(\theta_{ij}^k\right)^2\right] \\ &- \lambda_2 \sum_{k<k'}\sum_{i,j}\omega_{ij}\left|\theta_{ij}^k - \theta_{ij}^{k'}\right|, \end{aligned} \tag{1}$$

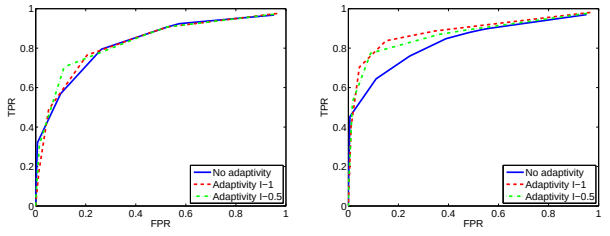
where $\theta_{ij}^k$ is the $ij$ element of $\Theta^k$; $\lambda_1$, $\lambda_2$ are the tuning parameters for the elastic net and fused penalties, respectively; and $\omega_{ij}$ are the adaptivity factors. We define the latter as follows

$$\omega_{ij} = \left[\sum_{k<k'}|\tilde{\theta}_{ij}^k - \tilde{\theta}_{ij}^{k'}| \sum_{k<k'}\sum_{l\in N_{ij}}\left(|\tilde{\theta}_{il}^k - \tilde{\theta}_{il}^{k'}| + |\tilde{\theta}_{jl}^k - \tilde{\theta}_{jl}^{k'}|\right)\right]^{-\gamma}, \tag{2}$$

where the $\tilde{\theta}_{ij}^k$ are the initial estimates of the network, $N_{ij}$ denotes the set of neighbors of link $(i, j)$, that is, the set of links connected to genes $i$ and $j$; and $\gamma$ is a positive parameter that controls the level of adaptivity. This adaptivity factor encourages fusion of link $(i, j)$

across all classes when they are already close, or when its neighbors are. By doing this, we adapt the tuning parameter for the fused penalty, $\lambda_2$, based on an initial zero-consistent estimate of the network ([7]).

Figure 1 shows the ROC curves, averaged over 50 replications, for modular and non-modular simulated data. They show a comparison of our method, Adaptivity I, for $\gamma = 1$ and $\gamma = 0.5$, and the regular fused lasso (No adaptivity). The figure shows that our method performs better in the presence of modularity and as well as the regular fused lasso when modularity is absent.




(a) Non-modular networks (b) Modular networks

Fig. 1: ROC curves. All methods perform similarly for non-modular networks. Adaptivity I performs better in presence of modularity.

**Generalization to specific pairwise penalties**
The adaptivity parameter (2) can be modified so it penalizes determined pairwise differences. This is necessary when looking for neighborhoods where all links are equal across the same subset of classes. In this case the adaptivity factor $\omega_{ij}^{kk'}$ has to be defined for each pair of classes $kk'$.

Data sets with unequal sample sizes present a challenge for network estimation. Classes with larger sample sizes tend to dominate the estimation in two ways. They make the estimated networks of classes with smaller sample sizes sparser than those with larger sample sizes. Also, they make classes with smaller sample sizes to fuse to each other faster than to classes with larger sample sizes. To alleviate this, we define a class specific sparsity penalty and a pairwise specific fused penalty, both corrected by an effective sample size [4].

Another interesting problem arises in the presence of ordered variables. For example, if survival data is available, samples can grouped in $T$ survival levels, thus creating a total of $KT$ classes. The fact that survival is an ordered variable implies that, for a given cancer class $k$, network links can be fused only for consecutive survival levels $t$ and $t+1$. This is part of our future work.

To optimize equation (1) in the presence of pairwise specific adaptivity factors $\omega_{ij}^{kk'}$, we solve the required fused lasso problem following [6].

We simulate data with pair-specific modular structure (Modular networks 2), as opposed example (Modular networks 1). In Figure 2 we show the box plots, computed over 100 replications, for the TPR and constrained FPR to approximately 0.1. No adaptivity corresponds to the regular fused lasso, Adaptivity I to the non-pairwise specific fused penalty with and Adaptivity II to the pairwise specific case, also computed with $\gamma = 1$.

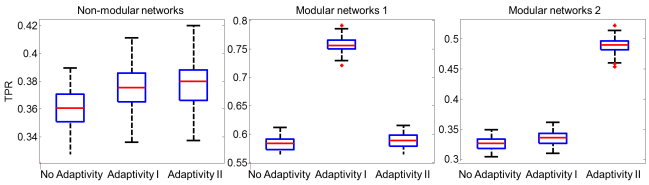



Fig. 2: Box plots for TPR with FPR=0.1. All methods perform similarly for non-modular networks. Our method performs better in presence of modularity.

We see that, in the absence of modularity all methods perform similarly. When the same modularity structure is present across all classes, Adaptivity I performs better. In the case of pair-specific class modularity Adaptivity II shows a superior performance.

**Acknowledgments**
We thank our collaborators from the Salhgrenska Academy Sven Nelander and Teresia Kling, and Holger Hoefling for sharing with us unpublished results.

## References

[1] Boyd, S. and Parikh, N. and Chu, E. and Peleato, B. and Eckstein, J. Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.

[2] Danaher, P., Wang, P. and Witten, D. The joint graphical lasso for inverse covariance estimation across multiple classes. *arXiv:1111.0324v1*, 2011.

[3] Jornsten, R. et al. Network modeling of the transcriptional effects of copy number aberrations in glioblastoma. *Molecular Systems Biology*, 7(486):485–516, 2011.

[4] Sánchez, J. and Kling, T. and Johansson, P. and Nelander, S. and Jörnsten, R. Comparative network analysis of human cancer: sparse graphical models with modular constraints and sample size correction. *Technical report, Chalmers University of Technology.*, 2013.

[5] Sharma, D. B. and Bondell, H. D. and Zhang, H. H. Consistent Group Identification and Variable Selection in Regression with Correlated Predictors. *Journal of Computational and Graphical Statistics* In press., 2012.

[6] Ye, G. and Xie, X. Split Bregman method for large scale fused Lasso. *Computational Statistics and Data Analysis.*, 55(4):1552–1569, 2011.

[7] Zou, H. The Adaptive Lasso and its Oracle Properties. *Journal of the American Statistical Association.*, 101(476):1418–1429, 2006.

# Learning Rates of $\ell_1-$regularized Kernel Regression

Lei Shi, Xiaolin Huang and Johan A.K. Suykens
Department of Electrical Engineering, KU Leuven,
ESAT-SCD-SISTA, B-3001 Leuven, Belgium
leishi@fudan.edu.cn (L. Shi), huangxl06@mails.tsinghua.edu.cn (X. Huang)
johan.suykens@esat.kuleuven.be (J. Suykens)

**Abstract**: We study the learning behavior of $\ell_1-$regularized kernel regression. We show that a function space involved in the error analysis induced by the $\ell_1$-regularizer and kernel function has nice behaviors in terms of the $\ell_2$-empirical covering numbers of its unit ball. Based on this result, we obtain the best learning rates of the algorithm so far.

**Keywords**: Learning theory, $\ell_1$-regularization, $\ell_2$-empirical covering number, error analysis

## 1 Introduction

The *regression* problem aims at estimating the function relations from random samples and occurs in various statistical inference applications. An output estimator of regression algorithms is usually expressed as a linear combination of features, i.e., a collection of candidate functions. As an important issue in statistical leaning theory and methodologies, *sparsity* focuses on studying the sparse representations of such linear combinations resulting from the algorithms. It is widely known that an ideal way to obtain the sparsest representations is to penalize the combinatorial coefficients by the $\ell_0-$norm. However, the algorithms based on $\ell_0-$norm often lead to an NP-hard discrete optimization problem, which motives the researchers to consider the $\ell_q-$norm ($0 < q \leq 1$) as the substitution. In particular, the $\ell_1-$norm constrained or penalized algorithms have achieved great success in a wide range of areas from signal recovery to variable selection in statistics. Due to the intensive study on compressed sensing, the algorithms involving the $\ell_1-$norm have drawn much attention in the last few years and been used for various applications, including image denoising, medical reconstruction and database updating.

Here, we focus on the $\ell_1-$regularized kernel regression. This algorithm minimizes a least-square loss functional adding a coefficient-based $\ell_1-$penalty term over a linear span of features generated by some kernel function. We establish a rigorous mathematical analysis on the asymptotic behavior of the algorithms under the framework of statistical learning theory.

## 2 Formal Setting

Let $X$ be a compact subset of $\mathbb{R}^d$ and $Y \subset \mathbb{R}$, $\rho$ be a Borel probability distribution on $Z = X \times Y$. For $f : X \to Y$ and $(x, y) \in Z$, the *least-square loss* is given by $(f(x) - y)^2$. Then the resulting target function is called *regression function* and satisfies

$$f_\rho = \arg\min \left\{ \int_Z (f(x) - y)^2 d\rho \middle| f : X \to Y, \text{measurable} \right\}.$$

In the supervised learning framework, $\rho$ is unknown and one estimates $f_\rho$ based on a set of observations $\mathbf{z} = \{(x_i, y_i)\}_{i=1}^m \in Z^m$ which is assumed to be drawn independently according to $\rho$. We additionally suppose that the conditional distribution $\rho(\cdot|x)$ is supported on $[-M, M]$, for some $M \geq 1$ and each $x \in X$.

Given a kernel function $K : X \times X \to \mathbb{R}$, the output estimator of $\ell_1-$regularized kernel regression is expressed as $\hat{f} = \sum_{i=1}^m c_i^{\mathbf{z}} K(x, x_i)$, where its coefficient sequence $\mathbf{c}^{\mathbf{z}} = (c_i^{\mathbf{z}})_{i=1}^m$ is a solution of the optimization problem

$$\min_{\mathbf{c} \in \mathbb{R}^m} \left\{ \frac{1}{m} \sum_{j=1}^m \left( y_j - \sum_{i=1}^m c_i K(x_j, x_i) \right)^2 + \gamma \|\mathbf{c}\|_1 \right\}.$$

Here $\gamma > 0$ is called a *regularization parameter* and $\|\mathbf{c}\|_1$ denotes the $\ell_1-$norm of $\mathbf{c}$. Recall that for any sequence $\mathbf{w} = (w_n)_{n=1}^\infty$, the $\ell_1-$norm is defined as

$$\|\mathbf{w}\|_1 = \sum_{n \in \text{supp}(\mathbf{w})} |w_n|,$$

where $\text{supp}(\mathbf{w}) := \{n \in \mathbb{N} : w_n \neq 0\}$. The kernel $K$ here is not necessarily symmetric or positive semi-definite, which leads to much flexibility.

## 3 Main Results

### 3.1 Capacity of the hypothesis space under $\ell_1-$constraint

We shall show a nice feature of the $\ell_1$-regularizer on tight bounds for $\ell_2$-empirical covering numbers of a related function space $\mathcal{H}_1$.

**Definition 1.** *Define a Banach space* $\mathcal{H}_1 = \left\{f : f = \sum_{j=1}^{\infty} \alpha_j K_{u_j}, \{\alpha_j\} \in \ell^1, \{u_j\} \subset X\right\}$ *with the norm*

$$\|f\| = \inf\left\{\sum_{j=1}^{\infty} |\alpha_j| : f = \sum_{j=1}^{\infty} \alpha_j K_{u_j}\right\}.$$

The continuity of $K$ ensures that $\mathcal{H}_1$ consists of continuous functions. Denote the ball of radius $R > 0$ as $\mathscr{B}_R = \{f \in \mathcal{H}_1 : \|f\| \leq R\}$.

The $\ell_2-$empirical covering number is defined by means of the normalized $\ell_2$-metric $d_2$ on the Euclidian space $\mathbb{R}^l$, i.e, for any $\mathbf{a} = (a_i)_{i=1}^l, \mathbf{b} = (b_i)_{i=1}^l \in \mathbb{R}^l$, the $\ell_2$-metric is given by

$$d_2(\mathbf{a}, \mathbf{b}) = \left(\frac{1}{l}\sum_{i=1}^{l} |a_i - b_i|^2\right)^{1/2}.$$

**Definition 2.** *For a subset $S$ of a pseudo-metric space $(\mathcal{M}, d)$ and $\epsilon > 0$, the covering number $\mathscr{N}(S, \epsilon, d)$ is defined to be the minimal number of balls of radius $\epsilon$ whose union covers $S$. For a set $\mathscr{F}$ of functions on $X$ and $\epsilon > 0$, the $\ell_2$-empirical covering number of $\mathscr{F}$ is given by*

$$\mathscr{N}_2(\mathscr{F}, \epsilon) = \sup_{l \in \mathbb{N}} \sup_{\mathbf{u} \in X^l} \mathscr{N}(\mathscr{F}|_{\mathbf{u}}, \epsilon, d_2),$$

*where for $l \in \mathbb{N}$ and $\mathbf{u} = (u_i)_{i=1}^l \subset X^l$, we denote the covering number of the subset $\mathscr{F}|_{\mathbf{u}} = \{(f(u_i))_{i=1}^l : f \in \mathscr{F}\}$ of the metric space $(\mathbb{R}^l, d_2)$ as $\mathscr{N}(\mathscr{F}|_{\mathbf{u}}, \varepsilon, d_2)$.*

We will present a bound on the logarithmic $\ell_2$-empirical covering numbers of $\mathscr{B}_1$. This bound holds for a general class of input space satisfying an *interior cone condition.*

**Definition 3.** *A subset $X$ of $\mathbb{R}^d$ is said to satisfy an interior cone condition if there exist an angle $\theta \in (0, \pi/2)$, a radius $R_X > 0$, and a unit vector $\xi(x)$ for every $x \in X$ such that the cone*

$$C(x, \xi(x), \theta, R_X) = \Big\{x + ty : y \in \mathbb{R}^d, |y| = 1, \\ y^T \xi(x) \geq \cos\theta, 0 \leq t \leq R_X\Big\}$$

*is contained in $X$.*

**Remark 1.** *The interior cone condition excludes those sets $X$ with cusps. It is valid for any convex subset of $\mathbb{R}^d$ with Lipschitz boundary [1].*

Now we are in the position to give our result on the capacity of $\mathscr{B}_1$.

**Theorem 1.** *Let $X$ be a compact subset of $\mathbb{R}^d$. Suppose that $X$ satisfies an interior cone condition and $K \in \mathscr{C}^s(X \times X)$ with $s \geq 2$ . Then there exists a constant $C_{X,K}$ that depends on $X$ and $K$ only, such that*

$$\log \mathscr{N}_2(\mathscr{B}_1, \epsilon) \leq C_{X,K} \epsilon^{-\frac{2d}{d+2\lfloor s \rfloor}} \log\left(\frac{2}{\epsilon}\right), \forall\ 0 < \epsilon \leq 1,$$

*where $\lfloor s \rfloor$ denotes the integral part of $s$.*

### 3.2 Convergence rates of $\ell_1-$regularized kernel regression

Due to the least-square nature, the performance of the algorithm can be measured by the error $\|\hat{f} - f_\rho\|^2_{L^2_{\rho_X}}$, where $\rho_X$ is the marginal distribution of $\rho$ on $X$ [3]. We say that $K$ is a *Mercer kernel* if it is continuous, symmetric and positive semi-definite on $X \times X$. Such a kernel can generate a *reproducing kernel Hilbert space* (RKHS) $\mathscr{H}_K$ [2]. For a continuous kernel function $K$, define

$$\widetilde{K}(u, v) = \int_X K(u, x) K(v, x) d\rho_X(x). \qquad (1)$$

Then one can verify that $\widetilde{K}$ is a Mercer kernel.

The following result is stated in terms of properties of the input space $X$, the measure $\rho$ and the kernel $K$.

**Theorem 2.** *Assume that $X$ is a compact convex subset of $\mathbb{R}^d$ with Lipschitz boundary, $K \in \mathscr{C}^s(X \times X)$ with $s > 0$ and $f_\rho \in \mathscr{H}_{\widetilde{K}}$ with $\widetilde{K}$ defined by (1). Let $0 < \delta < 1$ and*

$$\Theta = \begin{cases} \frac{d+2s}{2d+2s}, & \text{if } 0 < s < 1, \\ \frac{d+2\lfloor s \rfloor}{2d+2\lfloor s \rfloor}, & \text{otherwise,} \end{cases} \qquad (2)$$

*where $\lfloor s \rfloor$ denotes the integral part of $s$. Take $\gamma = m^{\epsilon - \Theta}$ with $0 < \epsilon \leq \Theta - \frac{1}{2}$. Then with confidence $1 - \delta$, there holds*

$$\|\hat{f} - f_\rho\|^2_{L^2_{\rho_X}} \leq C_\epsilon \log(6/\delta) \left(\log(2/\delta) + 1\right)^6 m^{\epsilon - \Theta}, \quad (3)$$

*where $C_\epsilon > 0$ is a constant independent of $m$ or $\delta$.*

**Acknowledgments**

Supported by GOA/10/09 MaNet, CoE EF/05/006 (OPTEC), FWO: G0226.06, G.0302.07, G.0588.09, SBO POM, IUAP P6/04 (DYSCO, 2007-2011), ERC AdG A-DATADRIVE-B, FWO structured modeling.

## References

[1] R. Adams and J. Fournier, Sobolev Spaces, Academic press, 2003.

[2] N. Aronszajn, Theory of reproducing kernels, Trans. Amer. Math. Soc. **68** (1950), 337–404.

[3] F. Cucker and D. X. Zhou, Learning Theory: An Approximation Theory Viewpoint, Cambridge University Press, 2007.

# Reduced Fixed-Size LSSVM for Large Scale Data

Raghvendra Mall
KU Leuven, ESAT-SCD
Kasteelpark Arenberg 10
B-3001 Leuven, Belgium
raghvendra.mall@esat.kuleuven.be

Johan A.K. Suykens
KU Leuven, ESAT-SCD
Kasteelpark Arenberg 10
B-3001 Leuven, Belgium
johan.suykens@esat.kuleuven.be

**Abstract**: We present sparse reductions to Fixed-Size LSSVM (FS-LSSVM) which can handle large scale data. The FS-LSSVM model achieves sparsity by solving the problem in the primal using a Nyström approximated feature map. To create this feature map a set of prototype vectors (PV) are selected. However, this solution is not the sparsest. We investigate the sparsity-error trade-off by introducing a second level of sparsity. This is done by means of iterative sparsifying $L_0$-norm based reduction on the FS-LSSVM model. We conduct experiments on two large scale real world datasets - Forest Covertype for classification and Year Prediction for regression to show the effectiveness of the proposed models and trade-off between sparsity and error estimations.

**Keywords**: sparse models, FS-LSSVM, $L_0$-norm

.

## 1 Introduction

The LSSVM model [1] has become a state-of-the-art technique in classification and regression. The LSSVM model solves an optimization problem which in the dual leads to solving a system of linear equations. A drawback of the LSSVM model is that each data point becomes a support vector(SV). Several works in literature including [2–8] address the problem of sparsity in the LSSVM model. But these techniques cannot guarantee a great reduction in the number of support vectors. One approach that proposed to directly enforce sparsity from the beginning was introduced in [9, 11] and is referred as the *fixed-size least squares support vector machines*. This method uses an explicit expression for the feature map using the Nyström method [10] using a set of prototype vectors (PV) of cardinality $M \ll N$. However, this is not the sparsest solution and choice of optimal value of $M$ is an open problem. In [11], they select $M$ as the number of PV required for the FS-LSSVM performance to be similar to the LSSVM performance.

In recent years, the $L_0$-norm has received increased attention. The $L_0$-norm counts the number of non-zero elements of a vector. So, when minimized it can result in extremely sparse models. But since it is a NP-hard problem several approximations to it have been discussed in [12, 14]. In this paper, we propose sparse reductions to a FS-LSSVM model using the iterative sparsifying procedure for $L_0$-norm introduced in [14]. The major motivation of obtaining a sparse solution is that sparseness allows memory and computationally efficient techniques and reduces the number of support vectors to decreases the out-of-sample prediction time.

## 2 Sparse Reductions to FS-LSSVM

### 2.1 $ALL_L_0$-norm FS-LSSVM model

For this method, we first need to build a FS-LSSVM model. For this purpose, we select $M$ prototype vectors by maximizing the quadratic Rènyi entropy which approximates the information in the large $N \times N$ kernel matrix with a smaller $M \times M$ kernel matrix. We then generate an explicit approximate feature map using Nyström approximation and solve the optimization problem in the primal resulting in the model $(\tilde{w}, \tilde{b})$ and the $L_0$-norm problem can be formulated as:

$$\begin{aligned} \min_{\tilde{w},\tilde{b},\tilde{e}} \quad & \mathcal{J}(\tilde{w},\tilde{e}) = \|\tilde{w}\|_0 + \frac{\gamma}{2}\sum_{i=1}^{N}\tilde{e}_i^2 \\ \text{s.t.} \quad & \tilde{w}^\intercal\tilde{\phi}(x_i) + \tilde{b} = y_i - \tilde{e}_i, i = 1,\ldots,N. \end{aligned} \tag{1}$$

The weight vector $\tilde{w}$ can be approximated as a linear combination of the $M$ prototype vectors i.e. $\tilde{w} \approx \sum_{j=1}^{M}\tilde{\beta}_j\tilde{\phi}(x_j)$ [13] where $\tilde{\beta}_j \in \mathbb{R}$ and need not be the Lagrange multipliers. We apply the regularization weight $\lambda_j$ on each of these $\tilde{\beta}_j$ to iteratively sparsify an approximate $L_0$-norm solution. Based on [14], we construct the following generalized primal problem:

$$\begin{aligned} \min_{\tilde{\beta},\tilde{b},\tilde{e}} \quad & J(\tilde{\beta},\tilde{e}) = \frac{1}{2}\sum_{j=1}^{M}\lambda_j\tilde{\beta}_j^2 + \frac{\gamma}{2}\sum_{i}^{N}\tilde{e}_i^2 \\ \text{s.t.} \quad & \sum_{j=1}^{M}\tilde{\beta}_j\tilde{K}_{ij} + \tilde{b} = y_i - \tilde{e}_i, i = 1,\ldots,N \end{aligned} \tag{2}$$

where $\tilde{K}$ is the kernel matrix such that $\tilde{K}_{ij} = \tilde{\phi}(x_i)^\intercal \tilde{\phi}(x_j)$, $x_i$ belongs to training set and $x_j$ belongs to the protoype vectors PV set. After elimination of $\tilde{e}_i$, the optimization problem can be re-written as:

$$\min_{\tilde{\beta},\tilde{b}} \quad J(\tilde{\beta},\tilde{b}) = \frac{1}{2}\sum_{j}^{M}\lambda_j\tilde{\beta}_j^2 + \frac{\gamma}{2}\sum_{i}^{N}(y_i - (\sum_{j}^{M}\tilde{\beta}_j\tilde{K}_{ij} + \tilde{b}))^2. \tag{3}$$

The equation (3) allows extending this approach for large scale datasets. The solution to (3) can be obtained by solving:

$$\left[\begin{array}{c|c} \tilde{K}^\intercal\tilde{K} + \frac{1}{\gamma}diag(\lambda) & \tilde{K}^\intercal 1_N \\ \hline 1_N^\intercal\tilde{K} & 1_N^\intercal 1_N \end{array}\right]\left[\begin{array}{c}\tilde{\beta}\\ \hline \tilde{b}\end{array}\right] = \left[\begin{array}{c}\tilde{K}^\intercal y\\ \hline 1_N^\intercal y\end{array}\right] \tag{4}$$

where $\tilde{K}^\intercal\tilde{K}$ is a $M \times M$ matrix. The procedure to obtain sparseness involves iteratively solving the system (4) for decreasing values of $\lambda$ as shown in [14]. To prevent the $L_0$-norm approximation from falling in a bad local minima, we initialize $\tilde{\beta}_j = \tilde{w}_j, j = 1,\ldots,M$. For each iteration the $\lambda_j = \frac{1}{\tilde{\beta}_j^2}$. Once we obtain the reduced set (SV) of support vectors, we re-perform FS-LSSVM using this SV set. More details about the proposed method can be found in [15].

### 2.2 Experiments

We conducted experiments on two large scale datasets with nearly 0.5 million points each. Results are depicted in Table 1. We observe that the number of support vectors reduce without much difference in the error estimations.

| | Forest Cover | | | Year Prediction | | |
|---|---|---|---|---|---|---|
| Algorithm | Error | SV | Time | Error | SV | Time |
| FS-LSSVM | $0.2 \pm 0.02$ | 763 | $118500 \pm 9294$ | $0.4 \pm 0.03$ | 718 | $248046 \pm 15330$ |
| ALL-$L_0$-norm | $0.22 \pm 0.03$ | 505 | $118570 \pm 9248$ | $0.44 \pm 0.05$ | 595 | $248412 \pm 15248$ |

**Tab. 1:** Mean Test Performance for 10 randomizations

## 3 Conclusion

We proposed a sparse reduction to FS-LSSVM model for large scale data which results into sparser solution than the FS-LSSVM model without significant difference in error estimations.

### Acknowledgments

This work was supported by Research Council KUL, ERC AdG A-DATADRIVE-B, GOA/10/09MaNet, CoE EF/05/006, FWO G.0588.09, G.0377.12, SBO POM, IUAP P6/04 DYSCO, COST intelliCIS.

## References

[1] J. A. K. Suykens and J. Vandewalle. Least Squares Support Vector Machine Classifiers. *Neural Processing Letters*, 9(3):293-300, 1999.

[2] D. Geebelen, J. A. K. Suykens and J. Vandewalle. Reducing the Number of Support Vectors of SVM Classifiers Using the Smoothed Separable Case Approximation. *IEEE Transactions on Neural Networks and Learning Systems*, 23(4): 682-688, April 2012.

[3] C. J. C Burges. Simplified support vector decision rules. In Proceedings of $13^{th}$ International Conference on Machine Learning, 71-77, 1996.

[4] B. Schölkopf, S. Mika, C. J. C. Burges, P. Knirsch, K. Müller, G. Rätsch and A. J. Smola. Input space verus feature space in kernel-based methods". *IEEE Transactions on Neural Networks and Learning Systems*, 10(5): 1000-1017, September 1999.

[5] T. Downs, K. E. Gates and A. Masters. Exact Simplification of Support Vector Solutions. *Journal of Machine Learning Research*, 2:293-297, December 2001.

[6] Y. J. Lee and O. L. Mangasarian. RSVM: Reduced Support Vector Machines. In Proceedings of the $1^{st}$ SIAM International Conference on Data Mining, 2001.

[7] J. A. K. Suykens, L. Lukas and J. Vandewalle. Sparse approximation using Least Squares Support Vector Machines. In Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS 2000), 757-760, 2000.

[8] Y. Li, C. Lin and W. Zhang. Improved Sparse Least-Squares Support Vector Machine Classifiers. *Neurocomputing*, 69(13): 1655-1658, 2006.

[9] J. A. K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor and J. Vandewalle. Least Squares Support Vector Machines, World Scientific Publishing Co, Pte, Ltd (Singapore), (ISBN: 981-238-151-1), 2002.

[10] C. K. I. Williams and M. Seeger. Using the Nyström method to speed up kernel machines. *Advances in Neural Information Processing Systems*, 13:682-688, 2001.

[11] K. De Brabanter, J. De Brabanter, J. A. K. Suykens and Bart De Moor. Optimized Fixed-Size Kernel Models for Large Data Sets. *Computational Statistics & Data Analysis*, 54(6): 1484-1504, 2010.

[12] E. J. Candes, M. B. Wakin and S. Boyd. Enhancing Sparsity by Reweighted l1 Minimization. *Journal of Fourier Analysis and Applications*, 14(5): 877-905, Special issue on sparsity, 2008.

[13] G. C. Cawley and N. L. C. Talbot. Improved sparse least-squares support vector machines. *Neurocomputing*, 48(1-4): 1025-1031, 2002.

[14] K. Huang, D. Zheng, J. Sun, Y. Hotta, K. Fujimoto and S. Naoi. Sparse Learning for Support Vector Classification. Pattern Recognition Letters, 31(13):1944-1951, 2010.

[15] M. Raghvendra and J. A. K. Suykens. L0-reduced LSSVMs for Large Scale Data. Internal Report 12-185, ESAT-SISTA, K.U. Leuven (Leuven, Belgium), 2012.

## A Notations

| | **FS-LSSVM** | | $L_0$**-norm** | | **ALL-$L_0$-norm FS-LSSVM** |
|---|---|---|---|---|---|
| SV/Train | $N/N$ | | $M/N$ | | $M'/N$ |
| Primal | $\boxed{\tilde{w}}$ | Step 1 | $\boxed{\tilde{\beta}}$ | Step 2 | $\boxed{w'}$ |
| | $\boxed{\tilde{\phi}:\mathbb{R}^d \rightarrow \mathbb{R}^M}$ | $\rightarrow$ | $\boxed{\tilde{K} \in \mathbb{R}^{N\times M}}$ | $\rightarrow$ | $\boxed{\phi':\mathbb{R}^d \rightarrow \mathbb{R}^{M'}}$ |

**Tab. 2:** Related to the sparse reduction of FS-LSSVM method. First FS-LSSVM is performed in the Primal. In Step 1, an iterative sparsifying $L_0$-norm procedure is performed in Primal where $\tilde{w} \approx \sum_{i=1}^{M}\tilde{\beta}\tilde{\phi}(x_i)$ and regularization weights $\lambda_j$ are applied on $\tilde{\beta}$. The $\tilde{K}$ matrix incorporates information about the entire training set ($N$). Finally, FS-LSSVM is re-performed in the Primal using the reduced SV set ($M'$).

# Pattern Recognition for Neuroimaging Toolbox

Jessica Schrouff
University of Liège, Belgium
jschrouff@ulg.ac.be

Maria J. Rosa
University College London, UK
m.rosa@ucl.ac.uk

Jane Rondina
University College London, UK

Andre F. Marquand
Kings College, London, UK

Carlton Chu
NIMH, NIH, USA

John Ashburner
University College London, UK

Jonas Richiardi
Stanford University, USA

Christophe Phillips
University of Liège

Janaina Mourao-Miranda
University College London, UK

**Abstract**: In the past years, mass univariate statistical analyses of neuroimaging data have been complemented by the use of multivariate pattern analyses, especially based on machine learning models. While these allow an increased sensitivity for the detection of spatially distributed effects compared to univariate techniques, they lack an established and accessible software framework. Here we introduce the "Pattern Recognition for Neuroimaging Toolbox" (PRoNTo), an open-source, cross-platform and MATLAB-based software comprising many necessary functionalities for machine learning modelling of neuroimaging data.

**Keywords**: software, neuroimaging, machine learning

## 1 Introduction

Various imaging modalities, such as functional/structural Magnetic Resonance Imaging (fMRI/sMRI) and Positron Emission Tomography (PET), have been developed to record brain structure and activity. Until recently, such data were analysed using standard univariate statistics, for example by linking the time-series of the signal in each voxel with a regressor, such as in the General Linear Model (GLM) implemented in Statistical Parametric Mapping (SPM, [2]). Although univariate analyses have proven powerful for making regionally specific inferences on brain function and structure, there are limitations to the type of research questions that they can address. More recently, these mass univariate analyses have been complemented by the use of pattern recognition analyses, in particular using machine learning based predictive models [3]. These analyses focus on predicting a variable of interest (e.g. mental state 1 vs. mental state 2, or patients vs. controls) from the pattern of brain activation/anatomy over a set of voxels. Due to their multivariate properties, these methods can achieve relatively greater sensitivity and are able to detect subtle, spatially distributed patterns in the brain. Potentially, pattern recognition can also be used to perform computer-aided diagnostic of neurologic or psychiatric disorders. Currently, the existing implementations consist of small code snippets, or sets of packages, and lack a dedicated single, integrated, and flexible software framework. In addition, the use of existing packages often requires high-level programming skills.

## 2 Pattern Recognition for Neuroimaging Toolbox

The "Pattern Recognition for Neuroimaging Toolbox" (PRoNTo[1], [6]) is a user-friendly and open-source toolbox that makes machine learning modelling available to every neuroimager. In PRoNTo, brain scans are treated as spatial patterns and learning models are used to identify statistical properties of the data that can be used to discriminate between experimental conditions or groups of subjects (classification models) or to predict a continuous measure (regression models). In terms of neuroimaging modalities, PRoNTo accepts NIfTI files[2] and can therefore be used to analyse sMRI and fMRI, PET, SPM contrast images and potentially any other modality in NIfTI file format. Its framework allows fully flexible machine learning based analyses and, while its use requires no programming skills, advanced users can easily access technical details and expand the toolbox with their own developed methods. Each step of the analysis can also be reviewed via user-friendly displays. Figure 1 provides an overview of the toolbox framework.

[1] PRoNTo, and all its documentation, are available to download freely from: `http://www.mlnl.cs.ucl.ac.uk/pronto/`

[2] `http://nifti.nimh.nih.gov/nifti-1/`

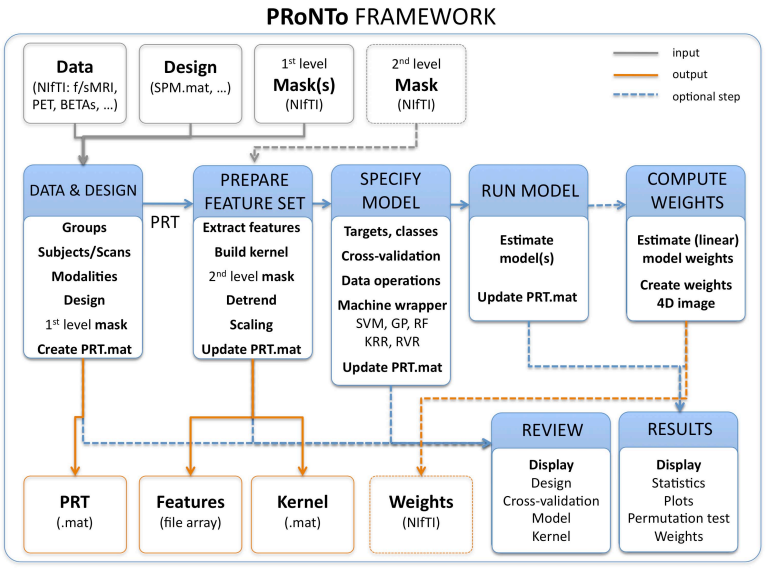



**Fig. 1:** PRoNTo framework. PRoNTo consists in five main analysis modules (blue boxes in the centre): dataset specification, feature set selection, model specification, model estimation and weights computation. In addition, it provides two main reviewing and displaying facilities (model, kernel and cross-validation displays, as well as, results display). PRoNTo receives as input any NIfTI image (comprising the data and a first-level mask, while an optional second-level mask can also be entered). The outputs of PRoNTo include: a data structure called PRT.mat, a data matrix (with all features), one or more kernels, and (optionally) images with the classifier weights.

PRoNTo can be used in three ways: through a graphical user-interface requiring no programming skills, using the MATLAB-batch system , or by scripting function calls. In the current version of PRoNTo, two linear kernel classification algorithms are embedded in the framework: Support Vector Machines ([1], LIBSVM implementation) and (binary and multiclass) Gaussian Process classification ([4], GPML toolbox). Regression can be performed using Kernel-Ridge Regression (KRR, [5]), Relevance Vector Regression (RVR, [7]) or Gaussian Processes Regression [4]. All algorithms are wrapped into what is called a "machine", which is independent from the design definition and cross-validation procedure. This allows an easy integration of new machine learning algorithms, enhancing the exchange of newly developed methods within the community, and the possibility to develop more advanced validation frameworks (e.g. nested cross-validation).

Several data sets were analysed with PRoNTo, showing the breadth of questions it can address [6]. As examples: Do patterns of brain activation recorded in fMRI encode information about a mental state? Can groups of subjects be distinguished based on features derived from their sMRI? Could their age be predicted with these neuro-anatomical features?

## 3 Discussion and Conclusions

In this work, we presented PRoNTo, a freely available software which addresses neuroscientific questions using machine learning based modelling. Although in its first version, PRoNTo provides both graphical interfaces for an easy use and a flexible programming framework. The authors therefore hope to facilitate the interaction between the neuroscientific and machine learning communities. On one hand, the machine learning community should be able to contribute to the toolbox with novel machine learning models. On the other hand, the toolbox should provide a variety of tools for the (clinical) neurosciencists, enabling them to ask new questions that cannot be easily investigated using existing statistical analysis tools.

**Acknowledgments**
The development of PRoNTo was mainly supported by the Pascal2 network and its Harvest Programme.

## References

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

[2] K. Friston, J. Ashburner, S. Kiebel, T. Nichols, and W. Penny. *Statistical Parametric Mapping: the analysis of functional brain images*. Elsevier Academic Press, London, 2007.

[3] T. Hastie, R. Tibshirani, and J. Friedman. *Elements of Statistical Learning*. Springer, 2003.

[4] C. Rasmussen and C. Williams. *Gaussian Processes for Machine Learning*. the MIT Press, 2006.

[5] C. Saunders, A. Gammerman, and V. Vovk. Ridge regression learning algorithm in dual variables. In *Proceedings of the 15th International Conference on Machine Learning*, pages 515–521, 1998.

[6] J. Schrouff, M. Rosa, J. Rondina, A. Marquand, C. Chu, J. Ashburner, C. Phillips, J. Richiardi, and J. M. ao Miranda. PRoNTo: Pattern Recognition for Neuroimaging Toolbox. *Neuroinformatics*, pages 1–19, 2013.

[7] M. Tipping. Sparse bayesian learning and the relevance vector machine. *Journal of Machine Learning Research*, 1:211–244, 2001.

# Stable LASSO for High-Dimensional Feature Selection through Proximal Optimization

Roman Zakharov, Pierre Dupont
Machine Learning Group, ICTEAM Institute,
Université catholique de Louvain, B-1348 Louvain-la-Neuve, Belgium
{roman.zakharov,pierre.dupont}@uclouvain.be

**Abstract**: The $l_1$-norm regularization is commonly used when estimating (generalized) linear models while enforcing sparsity. The automatic feature selection embedded in such an estimation is however known to be highly unstable since, among correlated features, an $l_1$ penalty tends to favor the selection of a single feature, essentially picked at random.
This paper introduces a modified optimization objective to stabilize LASSO or similar approaches. The solution to this modified problem is constrained by a norm ball rescaled according to the variances of the predictor variables. We further describe how such problems can be efficiently solved through proximal optimization.
Classification experiments conducted on several microarray datasets show the benefits of the proposed approach, both in terms of stability and predictive performances, as compared to the original LASSO, Elastic Net, Trace LASSO and a simple variance based filtering.

**Keywords**: feature selection, regularization, stability, LASSO, proximal optimization

## 1 Introduction

Feature selection aims at improving the interpretability of predictive models and at reducing the computational cost when predicting from new observations. Such a selection is also desirable when it is *a priori* known that the model should be sparse or to prevent overfitting. This is especially relevant when the number $p$ of input features, or predictor variables, largely exceeds the number $n$ of training observations. In such contexts, feature selection can also increase the predictive performances.

The $l_1$-norm is often used to regularize (generalized) linear models while performing an automatic feature selection by driving most model coefficients towards zero. The LASSO method [2] precisely combines such a regularization with a least square loss for estimating a regression model.

Predictive models estimated with a LASSO penalty are however known to be highly unstable, which means that small data perturbations can imply drastic changes in the subset of automatically selected variables. The lack of stability of the LASSO is generally attributed to the fact that, among several correlated features, an $l_1$ penalty tends to favor the selection of a single feature, essentially picked at random. In contrast, univariate filter methods, such as a $t$-test feature ranking, rely on general statistical characteristics of the data, which are much less sensitive to small data perturbations. Such simple selection methods are typically more stable but ignore the possible correlations between features and are not embedded into the estimation of a predictive model.

The S-LASSO method detailed in section 2 relies on a modified optimization objective to stabilize the LASSO. The solution to this modified problem is constrained by a norm ball rescaled according to the variances of the predictor variables. In contrast to the Elastic Net [5] and Trace LASSO [1] approaches, which favor the joined selection of correlated features, S-LASSO tends to discard low variance features because they are expected to be less informative.

## 2 A scaled proximal method for feature selection

Let $\mathbf{X} = (\mathbf{x_1}, \ldots, \mathbf{x_n})^T \in \mathbb{R}^{n \times p}$ be the design matrix made of $n$ training observations in $\mathbb{R}^p$, and $\mathbf{y} \in \mathbb{R}^n$ be the response vector. Learning the weight vector $\mathbf{w}$ of a simple linear model $y = \mathbf{w}^T\mathbf{x} + \varepsilon$, where $\varepsilon$ denotes a Gaussian noise with 0 mean and variance $\sigma^2$, is commonly phrased as a convex optimization problem of the form

$$\min_{\mathbf{w} \in \mathbb{R}^p} f(\mathbf{w}) + \Omega(\mathbf{w}), \tag{1}$$

where $f : \mathbb{R}^p \to \mathbb{R}$ is a convex differentiable loss function and $\Omega : \mathbb{R}^p \to R$ is a convex norm, not necessarily smooth or Euclidean. The $\Omega$ regularization term aims at reducing over-fitting by penalizing large absolute weight values, for which small input changes would have a significant impact on the predicted output.

In this work we propose to modify the general optimization problem (1) while rescaling the norm penalty according to individual feature variances:

$$\min_{\mathbf{w}\in R^p} f(\mathbf{w}) + \lambda \sum_{j=1}^{p} \frac{1}{r_j} \Omega(w_j), \qquad (2)$$

where vector $\mathbf{r} \in \mathbb{R}^p$ is proportional to the feature variances. We note that those variances are estimated *before* centering and normalizing the data to unit variance as usual when estimating a LASSO model. The proposed method also offers a general framework beyond this specific choice of variance weighting. In practice, any vector $r$ can be used to favor the selection of some variables a priori believed to be more relevant. This modified objective can be straightforwardly used with any penalty that can be decomposed component-wise. We focus on the $l_1$-norm as a special case of interest but we also report positive experimental results with an Elastic Net penalty. As compared to (1), the regularization ball associated to $\Omega(\mathbf{w})$ now takes the form of an ellipsoid elongated along the directions of higher variance. In other words, the regularization constant $\lambda$ is rescaled in a component-wise fashion.

The proposed method is related to the Adaptive LASSO [4] which also penalizes the $l_1$ penalty component-wise with adaptive weights. These weights are initially equal to the ordinary least square estimates and iteratively updated under the control of an additional tuning parameter. In contrast, S-LASSO does not require such an additional parameter and iterative reweighting as it relies on the observed variances along each dimension. The proposed framework is also not restricted to the $l_1$ penalty. Modification to the LASSO by some form of variance weighting has already been proposed in [3]. This related work describes bounds on the prediction error and oracle properties while we focus on the improved stability of the embedded feature selection as a result of such variance weighting. We also show how the S-LASSO modified objective can be efficiently solved through proximal optimization.

The modified objective (2) promotes stability since the identity of high variance features is expected not to change much while varying the data sampling. As shown experimentally, the predictive performances may also be improved since low variance features across learning observations are expected to be less informative for prediction. We also show that solving (2) offer better results than simply pre-filtering features based on their variances. We further detail how this modified objective can be efficiently solved through proximal optimization.

## 3 Results

We report experimental performances of the proposed method with a LASSO or Elastic Net penalty, and refer to those approaches as S-LASSO and S-ENET respectively. The competing approaches are the original LASSO or ENET with a logistic loss. We also report the performances obtained with TRACE LASSO adapted to a classification problem. Since S-LASSO and S-ENET use the individual feature variances to modify the optimization objective, we also compare to VARIANCE_RANKING, which is a filter method keeping only a desired number of features with the largest variances. Our experiments conducted on 5 microarray datasets illustrate the benefits of the proposed approach both in terms of stability of the gene selection and the classification performance, as compared to the original LASSO, ELASTIC NET or TRACE LASSO. In contrast, the stability of VARIANCE_RANKING is always very high but the predictive performances drop drastically when reducing the number of selected features.

**Acknowledgments**
Roman Zakharov is supported by a FRIA grant (5.1.191.10.F).

## References

[1] Edouard Grave, Guillaume Obozinski, and Francis Bach. Trace Lasso: a trace norm regularization for correlated designs. In *Advances in Neural Information Processing Systems*, Granada, Spain, 2011.

[2] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1994.

[3] Sara van de Geer, Peter Buehlmann, and Shuheng Zhu. The adaptive and the thresholded lasso for potentially misspecified models (and a lower bound for the lasso). *Electronic Journal of Statistics*, 5:688–749, 2011.

[4] Hui Zou. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, Vol. 101, No. 476, Theory and Methods:1418–1429, 2006.

[5] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society, Series B*, 67:301–320, 2005.

# Regularization in topology optimization

Atsushi Kawamoto, Tadayoshi Matsumori, Daisuke Murai and Tsuguo Kondoh
Toyota Central R&D Labs., Inc. Nagakute Aichi 480-1192, Japan
atskwmt@mosk.tytlabs.co.jp

**Abstract**: This paper deals with topology optimization in terms of the regularization of design spaces. Several filtering techniques have so far been proposed for regularizing design variables and restricting the minimum length scale of designs. They are categorized into two major groups, namely density filters and sensitivity filters. Both filters have their pros and cons in theory and practice. This paper is intended to revisit the theoretical background of topology optimization and discuss the long-lasting issue of regularization in topology optimization through the implementation of a PDE-based filter.

**Keywords**: topology optimization, regularization, filter

## 1 Introduction

Topology optimization is a design method that can optimize not only the sizes and dimensions but also shapes and topologies of target structures [1]. The idea of topology optimization is to consider structural designs as material distribution problems. The structural designs are represented by a scalar function called the characteristic function that takes on values between 0 and 1: 0 stands for void and 1 for solid. With this representation one can freely design complicated layouts of target structures by distributing materials. However, this representation may cause some numerical instabilities due to the lack of smoothness in the characteristic function. To alleviate the numerical instabilities, *image processing*-based filtering techniques are commonly used in continuum-based topology optimization. They are categorized into two major groups: density filters [2, 3] and sensitivity filters[6]. Sensitivity filters are applied to the design sensitivities rather than the design variables themselves. Among these filters the so-called convolution filter is the most widely used for removing the checkerboard patterns [1]. It is simple to implement and easy to understand its geometric effects. One drawback is, however, that there remain discrepancies between the filtered sensitivities and the actual sensitivities. On the other hand, PDE-based filtering techniques have also been proposed, in which Helmholtz type partial differential equations are solved [4, 5]. From the implementation point of view the PDE-based filters are advantageous because one can utilize the existing computational frameworks of FEM. In this paper, we explain the basic concept of topology optimization and the involved methodological elements, and then discuss the regularization of design spaces by specifically focusing on the application of the PDE-based filtering technique directly to the design variables instead of the design sensitivities.

## 2 Topology optimization

This section gives a short introduction to the topology optimization method and some inherent issues in the methodology. Figure 1 shows the design domain with the boundary and load conditions for a benchmark design problem, in which we maximize the stiffness of the so-called MBB beam by distributing a prescribed amount of material within the design domain. This problem can be formulated as the minimization of the mean compliance under the total volume constraint:

$$\min_{\rho\in[0,1]} f := \int_{\Gamma_{\mathrm{N}}} \mathbf{t}\cdot\mathbf{u}\,\mathrm{d}\Gamma \text{ s.t. } g := \int_{\mathrm{D}} \rho\,\mathrm{dD} - \overline{V} \leq 0. \quad (1)$$

Note that the design variables are relaxed so as to take intermediate values between 0 and 1. The displacement vector, $\mathbf{u}$, is calculated from the following analysis problem:

$$\left.\begin{array}{lll} -\nabla\cdot(\mathbf{E}:\boldsymbol{\epsilon}(\mathbf{u})) = \mathbf{0} & \text{in} & \mathrm{D} \\ \mathbf{u} = \mathbf{0} & \text{on} & \Gamma_{\mathrm{D}} \\ (\mathbf{E}:\boldsymbol{\epsilon}(\mathbf{u}))\cdot\mathbf{n} = \mathbf{t} & \text{on} & \Gamma_{\mathrm{N}} \end{array}\right\}, \quad (2)$$

where $\mathbf{E}$ stands for a homogeneous isotropic elasticity tensor and $\boldsymbol{\epsilon} := (\nabla\mathbf{u} + \nabla\mathbf{u}^{\top})/2$. The material density $\rho$ is embedded into the elasticity tensor as

$$\mathbf{E} = \rho^{P}\mathbf{E}_{\max} + (1-\rho^{P})\mathbf{E}_{\min}, \quad (3)$$

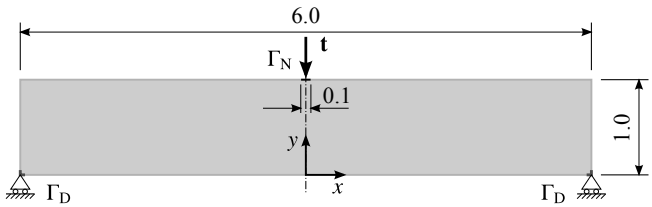



**Fig. 1:** Design domain and boundary conditions for the 2D minimum mean compliance problem



**Fig. 2:** Optimized structures for $P = 1$ (left) and $P = 3$ (right)

where $\mathbf{E}_{\max}$ is the elasticity tensor of the solid material, i.e.,$\mathbf{E} = \mathbf{E}_{\max}$ when $\rho = 1$; $\mathbf{E}_{\min}$ is the lower bound set for avoiding singularity when $\rho = 0$. $P(= 3, typically)$ is introduced to promote black and white solutions. The design domain is discretized by a square element with ength $\Delta x = 0.025$. In this problem we pursue a symmetric design. Therefore, we only solve the right half of the domain. This problem can be solved by a continuous gradient-based optimization method such as SNOPT. The results are shown in Figure 2. In the left structure a large gray area (grayscale problems) remains. By increasing the value of $P$ we can force the design variables toward 0 or 1. However, we typically end up with a solution like the right illustration of Figure 2 (checkerboard problems).

## 3 A PDE-based density filter

Since the density function, $\rho \in L^{\infty}(\mathrm{D})$, can take any point-wise value, the raw density function may produce severely oscillating designs like the checkerboard pattern shown in the right illustration of Figure 2. In order to regularize the density function we introduce the following partial differential equation for $\varrho \in H^1(\mathrm{D})$:

$$-R^2\nabla^2\varrho + \varrho = \rho. \tag{4}$$

Then we project $\varrho$ onto $\hat{\varrho}$ as

$$\hat{\varrho} = H(\varrho - 0.5; h), \tag{5}$$

where $H$ is a regularized (differentiable) Heaviside function with the half bandwidth $h$. By appropriately setting $R$ and $h$, these equations function like a low-pass filter that acts on the raw density function, $\rho$, to produce the smoothed density function, $\varrho$. Then we use the projected smooth density function, $\hat{\varrho}$, for the material interpolation in Eq. (3) instead.

## 4 Numerical examples

Figure 3 illustrates the optimized results for the density in grayscale when parameter $R$ in the PDE (4) is set to $\{1.0\Delta x, 2.0\Delta x, 4.0\Delta x\}$, and the half bandwidth, $h$, in the Heaviside projection (5) is set to $\{0.1, 0.5, 1.0\}$. The lesser $R$ and the tighter $h$ (the upper left) give finer details with a clear outline in the final design, while the opposite combination (lower right) makes the outline extremely blurred. Note that no checkerboard patterns appeared in all the cases even though the linear elements are used for the discretization.

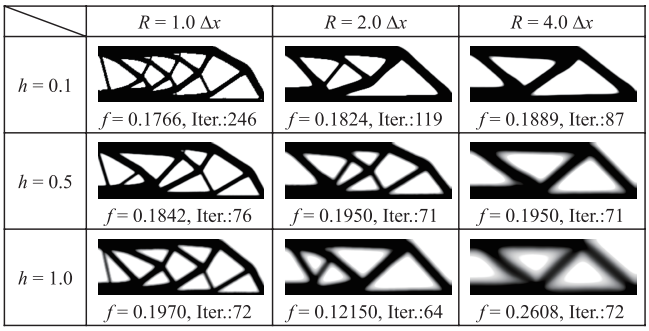



**Fig. 3:** Optimized results

## 5 Conclusions

We revisited the theoretical background of topology optimization and discussed the issues inherent in the regularization in topology optimization through the implementation of a PDE-based filter. Since the filter acts directly on the design variables, the consistency of the design sensitivities is preserved. The effectiveness of the proposed method is confirmed through numerical examples.

## References

[1] M.P. Bendsøe, O. Sigmund. *Topology Optimization – Theory, Methods, and Applications.* Springer Verlag: Berlin Heidelberg, 2003.

[2] B. Bourdin. Filters in topology optimization. *International Journal for Numerical Methods in Engineering*, 50(9):2143–2158, 2001.

[3] E.T. Bruns, D.A. Tortorelli. Topology optimization of non-linear elastic structures and compliant mechanisms. *Computer Methods in Applied Mechanics and Engineering*, 190(26-27): 3443–3459, 2001.

[4] A. Kawamoto, T. Matsumori, S. Yamasaki, T. Nomura, T. Kondoh, S. Nishiwaki. Heaviside projection based topology optimization by a pde-filtered scalar function. *Structural and Multidisciplinary Optimization*, 44:19–24, 2011.

[5] B.S. Lazarov, O. Sigmund. Filters in topology optimization based on Helmholtz-type differential equations. *International Journal for Numerical Methods in Engineering*, 86(6): 765–781, 2011.

[6] O. Sigmund. Morphology-based black and white filters for topology optimization. *Structural and Multidisciplinary Optimization*, 33(4-5): 401–424, 2007.

# Classification of MCI and AD patients combining PET data and psychological scores

F. Segovia, C. Bastin, E. Salmon and C. Phillips
Cyclotron Research Centre, University of Liège, Belgium
fsegovia@ulg.ac.be

**Abstract**: This study's aim was to measure the advantages of using psychological test data in the automatic classification of functional brain images in order to assist the diagnosis of neurodegenerative disorders such as Alzheimer's disease (AD). Several computer-aided diagnosis systems for AD based on PET images were developed. Some of them used psychological scores beside the image data in the classification step and others did not. The results show the ones that take into account the psychological scores achieve higher accuracy rates.

**Keywords**: Alzheimer's disease, machine learning, psychological scores

## 1 Introduction

In recent years, many computer-aided diagnosis (CAD) systems for neurodegenerative disorders have been presented [3, 5, 6]. Based on the assumption that pathological manifestations of these disorders appear some years before subjects become symptomatic [1, 8], they try to diagnose them even before the classical diagnosis procedure based on psychological tests does. These systems take advantage of the machine learning improvements carried out in the last decades [7] and report accuracy rates over 90% when trying to distinguish between patient and healthy controls [4].

Most of the recent CAD systems for neurodegenerative disorders are based on brain imaging (including MRI, fMRI, PET and SPECT data) and report high accuracy rates. The small sample size problem [2] can be addressed by means of a feature extraction technique that reduces the huge amount of data contained in a brain image into a relatively small unidimensional vector. In this case, the structure of the CAD systems based on neuroimaging and machine learning is as follows: After the preprocessing of the images (which involves the spatial registration and the intensity normalization), an algorithm is applied to select and summarize the relevant information. This information is rearranged in a vector and used as feature for the classification step. Finally, a statistical classifier is used to separate pathological and control subjects, performing that way the automatic diagnosis.

This paper shows that combining information from PET images and psychological tests can improve the accuracy of CAD systems for dementia. Specifically, we have developed several CAD systems to measure the benefit of using some psychological scores beside the image features in the classification step. The extraction of the image features was carried out by applying three different techniques, namely Principal Component Analysis (PCA), Partial Least Squares (PLS) and Independent Component Analysis (ICA). The results clearly show that using the psychological scores beside the image features improves the global accuracy of the systems.

## 2 Methods

Some dimension reduction techniques were applied to reduce the information contained in the brain images:

**Principal Component Analysis** (PCA) is a simple, non-parametric method of extracting relevant information from confusing data sets. Mathematically, PCA performs a linear transformation that transforms the data to a new coordinate system such that the largest variance by any projection of the data comes to lie on the first dimension, the second largest variance on the second dimension, and so on.

**Partial Least Squares** (PLS) is a statistical method similar to PCA, however PLS carries out the transformation by maximizing the covariance between the data and some properties of the data. For classification purposes the PLS1 variant, which considers the class belonging information as the unique property of the data, was used.

**Independent Component Analysis** (ICA) is a computational method for separating a multivariate signal into additive subcomponents supposing the mutual statistical independence of the non-Gaussian source signals. ICA has been successfully applied to dimension reduction problems by projecting the data into its independent components, performing that way the reduction.

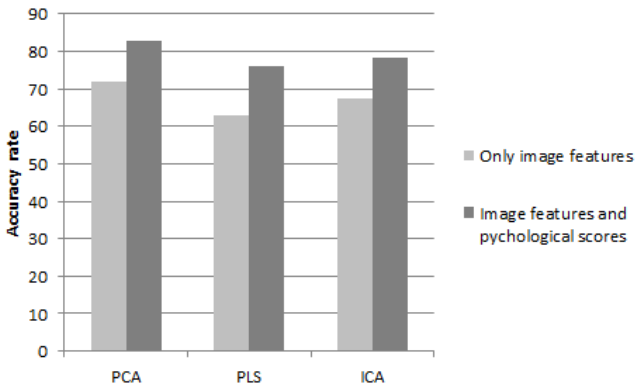



**Fig. 1:** Accuracy rates achieved by the developed CAD systems compared with the achieved ones by the same systems but using only the PET images as data source.

## 3 Experiments and results

Three different CAD systems were developed, each one implemented one of the three dimension reduction techniques mentioned in section 2. The final number of components (features) per image was determined by the percentage of variance (90% of the total) in the case of PCA and using the Fisher Discriminant Ratio in case of PLS and ICA (selected components gather 95% of total FDR). In addition, five psychological scores were used. Three of them were derived from a verbal cued recall memory task, reflecting respectively the efficiency of memory encoding (immediate recall), long-term episodic memory (cued recall) and monitoring capacities (intrusions). Two additional scores were phonemic (letter P) and semantic (animals) verbal fluency measures, as an index of executive functioning.

In this initial work, the combination of the image features and the psychological scores was carried out by concatenating both into a feature vector which size is the sum of the number of image features and psychological scores. The classification step was performed by means of a Support Vector Machine (SVM) classifier and a linear kernel. The accuracy of the systems was estimated using a database with 46 PET images from subjects originally diagnosed with Mild Cognitive Impairment (MCI). Those subjects whose diagnosis remained as MCI during three years or more were labeled as MCI and those whose diagnosis changed to dementia type Alzheimer in the same period were labeled AD. In summary, the database contains 20 MCI and 26 AD images and for each of them the 5 psychological scores described above. Because of the reduced number of images available, we used a leave-one-out cross-validation scheme to compute the performance of the systems. Figure 1 shows the accuracy rates obtained with both imaging features and psychological scores or with only the imaging features, for the 3 different dimension reduction techniques. When the psychological scores are included, the accuracy is higher than when only the image features are provided. The difference between these pairs of accuracy values were statistically assessed through a non-parametric test. 1000 sets of random psychological scores (same range as the original ones) were generated, then classifier was trained with these random scores (with the image features) and the accuracy estimated. A $p-$value was then calculated as the number of cases where the accuracy obtained with the random scores was larger than that obtained with the true scores, divided by 1000, i.e. the probability of obtaining a better accuracy with a random score. A $p-$value of 0.001, 0.002 and 0.01 was obtained for the PCA-, PLS- and ICA- based CAD systems respectively.

In light of these results we can conclude that adding some information from psychological tests to the automatic diagnosis system for neurodegenerative disorder does improve the accuracy of the systems.

## References

[1] Elisa Canu et al. Microstructural diffusion changes are independent of macrostructural volume loss in moderate to severe alzheimers disease. *The Journal of Alzheimer's Disease*, 19(3):963–976, 2010.

[2] R. P. W. Duin. Classifiers in almost empty spaces. In *Proceedings 15th International Conference on Pattern Recognition*, volume 2, pages 1–7. IEEE, 2000.

[3] Y. Fan et al. Spatial patterns of brain atrophy in MCI patients, identified via high-dimensional pattern classification, predict subsequent cognitive decline. *NeuroImage*, 39(4):1731–1743, 2008. PMID: 18053747.

[4] Manhua Liu et al. Hierarchical fusion of features and classifier decisions for alzheimer's disease diagnosis. *Human brain mapping*, February 2013. PMID: 23417832.

[5] Janaina Mouro-Miranda et al. Classifying brain states and determining the discriminating activation patterns: Support vector machine on functional MRI data. *NeuroImage*, 28(4):980–995, December 2005. PMID: 16275139.

[6] F. Segovia et al. A comparative study of feature extraction methods for the diagnosis of alzheimer's disease using the ADNI database. *Neurocomput.*, 75(1):6471, January 2012.

[7] Vladimir Vapnik. *The Nature of Statistical Learning Theory*. Springer, 2nd edition, November 1999.

[8] Jennifer L. Whitwell et al. 3D maps from multiple MRI illustrate changing atrophy patterns as subjects progress from mild cognitive impairment to Alzheimer's disease. *Brain*, 130(7):1777–1786, 2007.

# Kernels design for Internet traffic classification

Emmanuel Herbert, Stephane Senecal
Orange Labs
38-40 rue du Général Leclerc,
92130 Issy-les-Moulineaux, France
firstname.lastname@orange.com

Stephane Canu
LITIS EA 4108, INSA Rouen
Avenue de l'Université,
76800 Saint-Etienne-du-Rouvray, France
stephane.canu@insa-rouen.fr

**Abstract**: In the telecommunications industry, Internet Service Providers (ISP) are willing to provide secure access to the Internet. Thus, developing solutions for detecting malicious activities conducted on the deployed networks is a task of primary importance. Implementing these monitoring features requires to handle a large volume of traffic data. *Machine learning* based techniques are well suited to perform automatically this task. In this work, we take into account the non-numerical characteristics and the heterogeneousness of the data for this problem. We thus focus on *kernel methods*, and particularly on multiple kernel learning techniques. The main challenge is then to design ad hoc kernels adapted to perform traffic classification operations. Furthermore, such a multiple kernel method allows to achieve a comprehensive representation of the network data for easing malicious activities detection.

**Keywords**: Telecommunications, Internet, Machine Learning, Classification, Kernel Methods, Multiple Kernel Learning.

Important industrial challenges faced nowadays by telecommunications and especially Internet operators deal with ensuring security for their networks and their users. In this respect, being able to detect and to monitor malicious activities on the networks has become a crucial task for operational network deployment. Concerning the Internet, *Domain Name System (DNS)* is the service mapping a *Fully Qualified Domain Name* (*FQDN*, e.g. "www.google.com") to Internet Protocol (IP) addresses (e.g. "173.194.34.16"). The IP address is the machine-readable identity number for a computer connected to the Internet network, a host. A FQDN is the human-readable identity name for a host. DNS plays a central role in the Internet as network communications are performed using the FQDN. Therefore, considering the DNS traffic data is relevant for implementing malicious activities detection solutions on an Internet Service Provider's (ISP) network.

The volume of data to process in practice is very large: the DNS traffic can up to $40,000$ queries per second on an Internet resolving server. To monitor such a volume, ISP need to deploy methods to automatically aggregate and classify the data. The goal we pursue is to highlight suspicious DNS traffic, i.e. the small amount of traffic which announces malicious activities such as a botnet initiating a communication with its command and control server for instance. In this respect, we focus on FQDN queries on Internet servers, which are data of *heterogeneous* nature. Indeed, a FQDN query can be viewed as a string (i.e. "www.google.com"), indicating a location in a labeled tree data structure. The FQDN is also characterized by the lists of its related IP addresses and of the caching durations associated to the DNS mapping. We then seek for temporal patterns in the sets of FQDN queries in order to detect abnormal traffic activities. These are the variables of interest for our classification problem. Current solutions to perform malicious activities detection tasks but with slightly different variables are based on decision trees classification methods [1].

Because the data are heterogeneous, as they are composed of strings, positions in a labeled tree, numerical values and lists of numerical values (from one to several dozens of items for each FQDN), we naturally consider kernel based methods for implementing the classification task for the DNS traffic. *Kernel methods* are a set of powerful regression and classification techniques which are useful for pattern analysis of non-numerical data, see [2] for instance. We are motivated by the richness of the data representation provided by kernel methods compared to the classical representation based on decision trees. Moreover, it is easier to obtain a visual representation of data with tools provided by kernel methods than with decision trees. For the data aggregation and classification problem under consideration here, an interesting approach is to use *multiple kernel learning* techniques [3]. The core idea is to deal with a kernel composed of a linear combination of several kernels, each of them being applied to a homogeneous part of the heterogeneous dataset. An important aspect for implementing this technique is to tune the coefficients of this linear combination. This aspect is not investigated further in this abstract, we focus instead on the design of the kernels for the combination.

One of the design challenges to tackle is to define a kernel enabling the representation of the FQDN *proximities*, i.e. the representation of the FQDN closeness on the labeled tree and with respect to their associated parameters. For instance how close are "www.google.com" and "www.orange.com" as strings, as elements on a labeled tree of domains, from their queries timestamps, and with respect to their related lists of IP addresses and associated caching durations? To ensure a relevant representation for the FQDN, we recall that the application of a kernel provides a distance metric in a different representation space, namely the *feature space*, thanks to the derived scalar product in this space. Proximity in the feature space should reflect the closeness of FQDN data in their initial representation space.
The approach we investigate for strings is based on the kernel design techniques for document classification [4]. For instance, for strings it is possible to consider *all-spectrum kernels* [2], which measure the similarity between two strings based on their common $n$-grams. Thanks to these techniques, it is possible to design the kernel for strings representing FQDN.
Then, for the location in the domain name tree of a FQDN, we can measure the *length of the common suffix* between two FQDN considering the level domain as a character [2]. The related kernel is then based on the length of this common suffix. Instead of viewing the FQDN only as a string, i.e. a sequence of characters, it can also be handled as a sequence of level domains. It is thus possible to locate FQDN on the labeled tree.
Also, in order to design kernels for the numerical values lists characterizing a FQDN, it is interesting to consider these lists as sets with appearance frequencies associated to each element, as documents for information retrieval are handled as sets of words (*"bag-of-words"*). We then take into account similarities between items of the lists. For instance, semantic relations between words are considered for document processing. The method used to grasp similarities between items of the lists is based on the distance induced by the application of the kernel to the list elements.
For the IP addresses, the network proximity is characterized by the membership to a subnetwork and by the localization of this subnetwork on the Internet. The subnetwork membership is provided by the first digits of the IP address in its binary representation. The kernel for the IP addresses is then based on the *length of the common prefix* in this binary representation. For this kernel design, the longest the common prefix is, the closest the IP addresses considered are. As for the bag-of-words representation technique for documents involving the term proximity matrix, we consider and use here the Gram matrix of all possible IP addresses.
For the caching durations, as they are usual numerical values, classical kernels like *Gaussian or polynomial*, are relevant to represent the data.
For queries timestamps, we design the kernel similarly as in [5]. For a given FQDN, a first aggregation step consists in splitting the observation period into timeslots, the number of the queries are then counted in each timeslot. Finally, we compute the *autocorrelation* vector of this *time series*. This vector highlights periodical behaviors for queries. It is used to built the related kernel, as an element of the feature space. Moreover, we also consider queries frequencies instead of their number for a given timeslot. It is thus possible to detect unusual amounts of queries for a specific FQDN.
We ran numerical experimentations on a $10mn$ capture of raw traffic. The FQDN whitelist (resp. blacklist) consists in 500 (resp. 125) randomly chosen FQDN in Alexa top 1000 (resp. in an internal blacklist) and in the queried FQDN from the traffic capture. As a first step, we tested kernel based unsupervised (kernel PCA [6]) and supervised (kernel FDA) methods on this dataset.

The work presented in this extended abstract takes into account the non-numerical features and the heterogeneousness of the Internet DNS traffic data in comparison to existing implementations of Internet network malicious activities detection [1]. The approach proposed is based on multiple kernel learning method [3]. One of the challenges tackled by the solution investigated is to design efficiently the kernels for handling jointly strings, IP addresses lists and timestamps lists representing the FQDN queries. Once such kernels are built, it is possible to come up with a comprehensive representation of the Internet DNS traffic data. A next step of this work consists in exploiting fully the richness of this data representation by developping ad hoc and efficient classification and data aggregation methods dedicated to these kernels.

## References

[1] L. Bilge, E. Kirda, C. Kruegel and M. Balduzzi, *Exposure: Finding malicious domains using passive DNS analysis*, Proc. of NDSS, 2011.

[2] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*, Cambridge University Press, 2004.

[3] A. Rakotomamonjy, F. Bach, S. Canu and Y. Grandvalet, *SimpleMKL*, Journal of Machine Learning Research, 9:2491-2521, 2008.

[4] D. Gusfield, *Algorithms on strings, trees and sequences: computer science and computational biology*, Cambridge University Press, 1997.

[5] Z. Li, B. Ding, J. Han, R. Kays and P. Nye, *Mining periodic behaviors for moving objects*, Proc. of ACM SIGKDD, 2010.

[6] B. Scholkopf, A. Smola and K.-R. Muller, *Nonlinear component analysis as a kernel eigenvalue problem*, Neural Computation, 10(5):1299-1319, 1998.

# Kernel Adaptive Filtering: Which Technique to Choose in Practice

Steven Van Vaerenbergh
Dept. of Communications Engineering
University of Cantabria, Spain
steven@gtas.dicom.unican.es

Ignacio Santamaría
Dept. of Communications Engineering
University of Cantabria, Spain
nacho@gtas.dicom.unican.es

**Abstract**: The field of kernel adaptive filtering has produced a myriad of techniques throughout the past decade. While each algorithm provides some advantages over others in certain scenarios, it is often not clear which technique should be used on a practical problem in which specific restrictions are in place. We propose to assess the quality of the solution as a function of the required cost, which can be measured as computational complexity, required memory or convergence speed. The obtained figures of merit allow us to decide which algorithm to use in practice. We include results on several benchmark data sets.

**Keywords**: kernel adaptive filtering, online kernel methods, comparison, benchmarks

## 1 Machine learning meets adaptive filtering

Kernel adaptive filtering is the subfield of online kernel-based learning that deals with the problem of regression. Specifically, given an input-output stream of data pairs $(\mathbf{x}_t, y_t)$, the task consists in estimating the function $f_t(\cdot)$ that relates input and output,

$$y_t = f_t(\mathbf{x}_t), \tag{1}$$

and to update this estimate efficiently every time a new data pair is received. Notice that the unknown function $f_t(\cdot)$ may be changing over time, as opposed to standard regression problems that assume static underlying models $f(\cdot)$. This problem is encountered for instance in adaptive signal processing theory, which classically deals with linear techniques, i.e. assuming a solution of the form, $f_t(\mathbf{x}_t) = \mathbf{w}_t^\top \mathbf{x}_t$ [1]. Kernel adaptive filtering applies machine learning techniques in order to obtain general nonlinear solutions to the online regression problem. Apart from the online scenario in which data arrives sequentially, these techniques are also often applied to static regression problems when the amount of data is too large to fit in the memory.

## 2 Existing methods

A comprehensive introduction to kernel adaptive filtering can be found in [2]. Though kernel adaptive filtering techniques draw upon several interesting properties of kernel methods, they also present some bottlenecks that are typically encountered in online kernel-based learning. Specifically, they suffer from growing complexity, since the number of kernels required to represent their solution can grow linearly or faster with the number of processed data. Different techniques have been proposed throughout the past decade to deal with this problem. A second bottleneck is parameter selection, which is generally still considered to be an open problem. Some optimization techniques have been proposed to this end, though most methods rely on standard cross-validation or even rules of thumb.

The majority of techniques can be categorized into either least-mean squares techniques, which have linear complexity in terms of the stored data, or recursive least-squares based techniques, which have quadratic complexity. Two of the pioneering methods stemmed from research in Gaussian processes [3] and least-squares support vector machines [4]. A closer relationship to classical adaptive filtering was established on the one hand by the Naïve Online regularized Risk Minimization Algorithm (NORMA) [5], Kernel Least-Mean Squares [6], and KNLMS/KAP [7], and on the other hand by the Kernel Recursive Least Squares (KRLS) algorithm [8]. Up to this point, however, all algorithms except NORMA assumed a stationary model.

A second generation of techniques dealt more explicitly with non-stationarity in order to perform tracking. Several techniques were proposed, ranging from the more straightforward Sliding-Window KRLS (SW-KRLS) [9] to the more sophisticated Quantized KLMS (QKLMS) [10], Kernel Adaptive Projection Subgradient Method (KAPSM) [11] and KRLS Tracker (KRLS-T) [12].

## 3 Benchmark testing framework

The performance of kernel adaptive filtering algorithms can be measured according to several different criteria, such as regression error, convergence rate or complex-

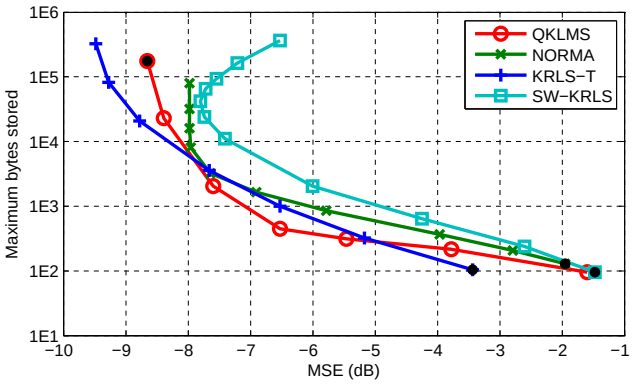



**Fig. 1:** Profile for memory usage versus prediction MSE, on the switching nonlinear channel data set. Each dot represents a single run of one of the algorithms, and a different parameter set was used for each dot.

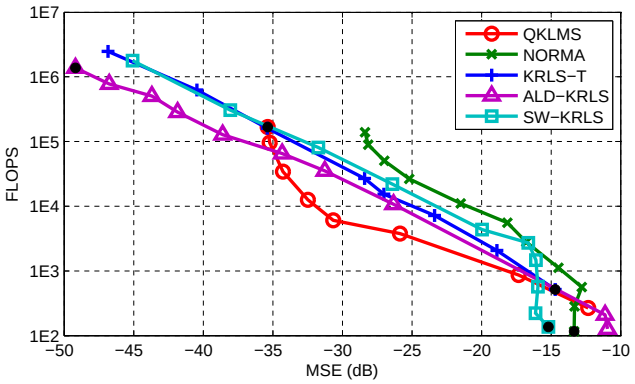



**Fig. 2:** Profile for average number of floating point operations versus prediction MSE, on the Mackey-Glass 30 set.

ity. In the literature it is common practice to compare different algorithms only according to one measure at a time. Nevertheless, an algorithm that scores well according to one criterion typically scores badly according to at least one other criterion. It is therefore often difficult for the practitioner to decide which method to use on a particular problem and data set.

We propose to represent the solution of an algorithm as a function of its cost. Depending on the application, cost measures of interest could be chosen out of computational complexity, required memory and speed of convergence. Two examples of trade-off figures are shown in Figs. 1 and 2. By fixing a performance goal, such as available memory or maximum allowable error, these figures allow us to determine which algorithm shows the most favorable remaining properties.

In order to make our results reproducible and extendable towards new algorithms, we have developed an open-source toolbox[1] that includes Matlab implementations of kernel adaptive filtering algorithms, and a profiler tool capable of generating trade-off figures.

## References

[1] A. H. Sayed, *Fundamentals of adaptive filtering*. Wiley-IEEE Press, 2003.

[2] W. Liu, J. C. Príncipe, and S. Haykin, *Kernel Adaptive Filtering: A Comprehensive Introduction*. Wiley, 2010.

[3] L. Csató and M. Opper, "Sparse representation for Gaussian process models," in *Advances in Neural Information Processing Systems 13*, pp. 444–450, MIT Press, 2001.

[4] A. Kuh, "Adaptive kernel methods for CDMA systems," in *International Joint Conference on Neural Networks (IJCNN 2001)*, vol. 4, pp. 2404–2409, IEEE, 2001.

[5] J. Kivinen, A. J. Smola, and R. C. Williamson, "Online learning with kernels," *IEEE Transactions on Signal Processing*, vol. 52, pp. 2165–2176, Aug. 2004.

[6] W. Liu, P. P. Pokharel, and J. C. Príncipe, "The kernel least-mean-square algorithm," *IEEE Transactions on Signal Processing*, vol. 56, no. 2, pp. 543–554, 2008.

[7] C. Richard, J. C. M. Bermudez, and P. Honeine, "Online prediction of time series data with kernels," *IEEE Transactions on Signal Processing*, vol. 57, pp. 1058–1067, Mar. 2009.

[8] Y. Engel, S. Mannor, and R. Meir, "The kernel recursive least squares algorithm," *IEEE Transactions on Signal Processing*, vol. 52, pp. 2275–2285, Aug. 2004.

[9] S. Van Vaerenbergh, J. Vía, and I. Santamaría, "A sliding-window kernel RLS algorithm and its application to nonlinear channel identification," in *2006 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 5, (Toulouse, France), pp. 789–792, May 2006.

[10] B. Chen, S. Zhao, P. Zhu, and J. C. Príncipe, "Quantized kernel least mean square algorithm," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, pp. 22–32, Jan. 2012.

[11] S. Theodoridis, K. Slavakis, and I. Yamada, "Adaptive learning in a world of projections," *IEEE Signal Processing Magazine*, vol. 28, pp. 97–123, Jan. 2011.

[12] S. Van Vaerenbergh, M. Lázaro-Gredilla, and I. Santamaría, "Kernel recursive least-squares tracker for time-varying regression," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, pp. 1313–1326, Aug. 2012.

[1] Available at `http://sourceforge.net/projects/kafbox/`

# Structured Machine Learning for Mapping Natural Language to Spatial Ontologies

Parisa Kordjamshidi
parisa.kordjamshidi@cs.kuleuven.be

Marie-Francine Moens
sien.moens@cs.kuleuven.be

**Abstract**: We propose a novel structured learning framework for mapping natural language to spatial ontologies. The applied spatial ontology contains spatial concepts, relations and their multiple semantic types based on qualitative spatial calculi models. To achieve a tractable structured learning model, we propose an efficient inference approach based on constraint optimization techniques. Particularly, we decompose the inference to subproblems, each of which is solved using LP-relaxation. This is done for both training-time and prediction-time inference. In this framework ontology components are learnt while taking into account the ontological constraints and linguistic dependencies among components. Particularly, we formulate complex relationships such as composed-of, is-a and mutual exclusivity during learning while the previous structured learning models in similar tasks do not go beyond hierarchical relationships. Our experimental results show that jointly learning the output components considering the above mentioned constraints and relationships improves the results compared to ignoring these. The application of the proposed learning model for mapping to ontologies is not limited to extraction of spatial semantics, it could be used to populate any ontology. We argue therefore that this work is an important step towards automatically describing text with semantic labels that form a structured ontological representation of the content.

**Keywords**: Spatial information extraction, Natural language semantics, Structured learning, Decomposed inference

## 1 Problem

**Motivation.** Extraction of spatial information from natural language is a challenging problem in applications such as robotics, geographical information systems and human-machine interaction. Here, we formulate a new machine learning task for extraction of spatial semantics considering spatial concepts and relations in sentences. We represent the semantics of the spatial relations based on qualitative spatial calculi models. The target ontology is represented in Figure 1. To deal with computational as well as semantic complexity of this task we assume two semantic layers in the proposed spatial ontology: a) spatial role labeling (SpRL) layer, b) spatial qualitative labeling (SpQL) layer.

**SpRL.** The SpRL layer considers cognitive-linguistic spatial roles including *spatial indicators*(ind), *trajectors*(tr) and *landmarks*(lm) and their relationships [1]. Spatial indicators indicate the existence of spatial information in a sentence. Trajector is an entity whose location is described and landmark is a reference object for describing the location of a trajector. Figure 2 shows an example sentence, *There is a white large statue with spread arms on a hill.*; In this example, the preposition *on* is a spatial indicator, *statue* is a trajector and its landmark is *hill*. A spatial relation is a triplet of spatial roles. There is one spatial relation in the mentioned sentence: $<on_{ind}\ statue_{tr} hill_{lm}>$. In general, there can be a number of spatial relations in each sentence. The spatial ontology is represented as a number of *single label*s that refer to one independent concept and *linked label*s that show the connection between the concepts in the ontology. For example, the spatial relation is a *linked label* that shows a *c*omposed-of relationship with the composing *single labels* of spatial roles.

**SpQL.** In the SpQL layer, the goal is to map the spatial triplets to a formal semantic representation. Spatial language, often conveys multiple meanings within one expression. Hence, our representation of the spatial semantics is based on multiple spatial calculi. Figure 1, shows the semantics that are considered in the SpQL layer. In fact, three general categories of regional (i.e topological), directional and distal relationships are considered and we allow multiple semantics to be assigned to each spatial triplet. The fine-grained topological relationships are based on the RCC-8 model including EC(externally connected), DC(disconnected), EQ(equal), PO(partially overlapping), PP(proper part). The directional relationships include relative directional relations of LEFT, RIGHT, FRONT, BEHIND, ABOVE, BELOW. For the distal information, a general class of DISTANCE is used in the applied spatial ontology. In Figure 2, the relation <on,statue,hill> is labeled with *regional* and $EC$ together with *directional* and $ABOVE$. For this sentence no distal information is annotated.

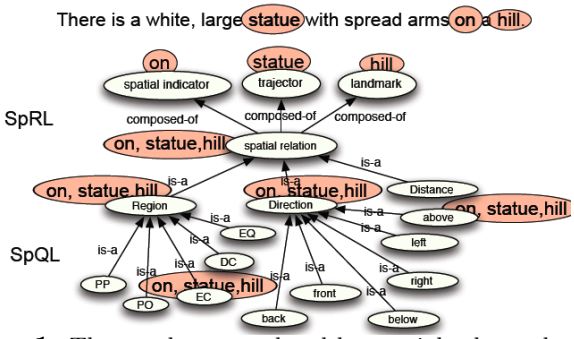



**Fig. 1:** The ontology populated by spatial roles and relations from the example sentence

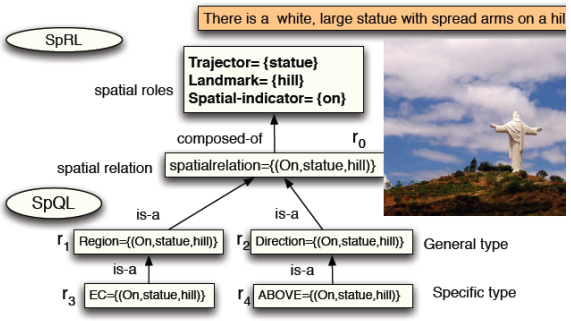



**Fig. 2:** Example of the CLEF-IAPR TC-12 benchmark

## 2 Structured learning model

We formulate the aforementioned problem as a supervised structured learning problem and solve it in the framework of structured support vector machines and structured perceptrons. In the supervised setting we learn a mapping $f : \mathcal{X} \rightarrow \mathcal{Y}$ between the input space $\mathcal{X}$ and discrete output space $\mathcal{Y}$ given a set of examples, $E = \{(X^{(i)}, Y^{(i)}) \in \mathcal{X} \times \mathcal{Y} : i = 1 \dots N\}$. In the structured learning, given the complex inputs and outputs, we learn an $F : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$ over input-output pairs. Then the prediction requires an inference task over $F$ to find the best $Y$ for a given input $X$. Therefore the general form of $f$ is : $f(X; W) = \arg\max_{Y \in \mathcal{Y}} F(X, Y; W)$. $F$ is assumed to be a linear function over a combination of input and output features $\Psi(X, Y)$, i.e., $F(X, Y; W) =< W, \Psi(X, Y) >$. The $X$s in our model are the natural language sentences that are composed of a number of *single components* such as words and *composed components* such as pairs and triplets of words. The $Y$s are the populated *single* and *linked labels* of the ontology with the relevant segments of the input sentence (see Figure 1). For a structured learning model, we need to design three main components [2]. The **first** component, is the **joint feature mapping** $\Psi(X, Y)$. The input features are defined based on the linguistically motivated features of the words in addition to the relations between adjacent and long distance words [1]. These features are joined with the spatial roles that each word plays in the output space and also with the semantic labels. The **second** component, is the **inference** algorithm. Since the number of possible $Y$'s for each input $X$ is very large in structured output prediction, the most violated $Y$ is considered in formulating the constraints for the structured learning optimization. However an efficient inference algorithm is required for finding the most violated $Y$ for a given $X$. To this aim we use the LP-relaxation technique in which we formulate the structural characteristics of the Y space in the form of hard constraints. We formulate the *is-a* and *composed-of* in the form of linear constraints. Moreover, the mutual exclusivity also should hold among the RCC and among the directional relations. By formulating the constraints we take into account the structure of the ontology during the training. We apply the same technique during the prediction to find the best ontological assignments for the segments of the input. However, solving a global LP-relaxation problem considering all variables in both layers is not tractable with the off-the-shelf solvers. Therefore we decompose the inference into subproblems based on the two semantic layers, each of which is solved using LP-relaxation. The subproblems should communicate to each other to achieve a global optimum. To apply this idea we use an approach similar to alternating optimization and we call it *communicative inference*. The **third** component, is the **loss** function. We use a component-based hamming loss as a part of the objective function during training. Such a loss function is linear; it is decomposed in terms of the output labels, hence easy to integrate in our model; moreover it is compatible with the semantics of our problem.

## 3 Experimental results

In the experiments we use the texts of CLEF IAPR TC-12 benchmark, containing 1213 sentences and 1706 annotated spatial relations. We implement a number of model variations using the structured support vector machines and structured perceptrons. The experimental results show that considering ontological relationships and constraints during training and prediction sharply improved the results in each layer compared to training local classifiers, in terms of F1-measure using 10-fold cross validation. We apply a number of relevant decomposed inference approaches as well as our proposed *communicative inference* during *training* and *prediction* for connecting the two layers. In our best results applying the proposed communicateve inference during *training* in our unified model improved the results of the SpRL layer but not the SpQL and applying it during *prediction* improved the results of both layers slightly compared to the pipelining of the two layers.

## References

[1] P. Kordjamshidi, M. Van Otterlo, and M.F. Moens. Spatial role labeling: Towards extraction of spatial relations from natural language. *ACM Trans. Speech Lang. Process.*, 8:1–36, December 2011.

[2] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6(2):1453–1484, 2006.

# Windowing strategies for on-line multiple kernel regression

Manuel Herrera
BATir - Université libre de Bruxelles
mherrera@ulb.ac.be

Rajan Filomeno Coelho
BATir - Université libre de Bruxelles
rfilomen@ulb.ac.be

**Abstract**: This work proposes two on-line learning multiple kernel regression (MKr) versions, to update the current model to a more accurate one, avoiding computational efforts associated with re-calculating the whole process each time that new data are available. The first approach is by sliding windows: strategy which maintains the size of the kernel matrix under study. The another one is by the so-called "worm" windows. It shrinks the kernel matrix as sliding windows does, but not at every entry of new data, attempting to lose a minimum of information.

**Keywords**: Multiple kernel regression, on-line learning, windowing methods

## 1 Introduction

Most of the kernel-based algorithms cannot be used to operate on-line since a number of difficulties such as time and memory complexities (due to the growing kernel matrix) and the need to avoid over-fitting. However, there are some works obtained in this sense during last years [1, 6, 7].

A kernel-based recursive least-squares algorithm that implements a fixed size "sliding-window" technique [5] has been proposed by Vaerenbergh *et al.*, 2006 [7]. We propose a similar methodology for resizing the kernel matrix to assist in the on-line process of multiple kernel regression (MKr) for mixed variables. The MKr process is summarized in the following two equations: Eq. 1 and Eq. 2. Where the kernel matrix used in the regression (Eq. 2) is calculated by a combination of kernels (Eq. 1).

$$\tilde{K}(x_i, x_j) = \sum_{s=1}^{M} \mu_s K_s(x_i, x_j) \tag{1}$$

$$\hat{f}(\boldsymbol{x}) = b + \sum_{i=1}^{n} (\alpha_i^+ - \alpha_i^-)\tilde{K}(x_i, \boldsymbol{x}) \tag{2}$$

The aim of the windowing strategies for on-line MKr (see Figure 1) is to improve the performance of the process without increasing the original algorithm's computation time. The new windowing strategy of "worm-windows" is proposed in this work. This method has two "expand-shrink" phases: firstly, it allows increasing the kernel matrix as its size remains adequate to work and there is not any over-fitting issue. Shrinking the kernel matrix to the original size is proposed when its size reaches to be not computationally efficient.

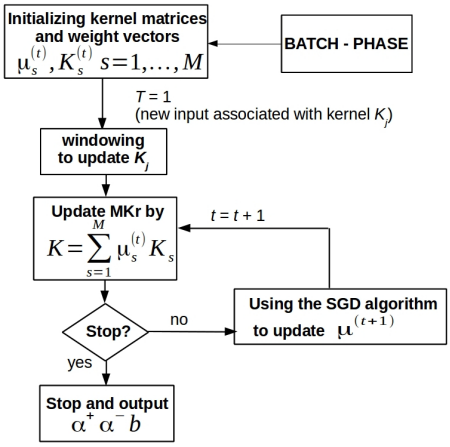



**Fig. 1:** Online multiple kernel regression.

## 2 Windowing for on-line MKr

### 2.1 On-line MKr by sliding windows

The sliding window approach consists in only taking the last $N$ pairs of the stream to performance the multi-kernel regression. When we obtain a new observed pair $\{x_{n+1}, y_{n+1}\}$, we first down-size the kernel matrix, $K_j^{(n)}$, by extracting the contribution from $x_{n-N}$ (see Eq. 3)

$$\check{K}_j^{(n)} = \begin{pmatrix} K_j^{(n)}(2,2) & \cdots & K_j^{(n)}(2,N) \\ \vdots & \ddots & \vdots \\ K_j^{(n)}(N,2) & \cdots & K_j^{(n)}(N,N) \end{pmatrix} \tag{3}$$

and then we augment again the $K_j^{(n)}$ dimension by importing the data input $x_{n+1}$ to obtain the kernel expressed in Equation 4.

$$K_j^{(n+1)} = \begin{pmatrix} \check{K}_j^{(n)} & K_j(\boldsymbol{X}_n, x_{n+1}) \\ K_j(x_{n+1}, \boldsymbol{X}_n) & K_j(x_{n+1}, x_{n+1}) + \lambda \end{pmatrix} \tag{4}$$

where $\boldsymbol{X}_n = (x_{n-N+1}, \ldots, x_n)^T$

Next, the kernel matrices are summed again (see Figure 1) and their weights, $\mu$, should be updated too. As it is a particular case of the calculation of weights corresponding to the batch phase of the overall process, the proposal is to follow an Stochastic Gradient Descent (SGD) [2, 3] algorithm.

### 2.2 On-line MKr by "worm" windows

The so-called *worm* window approach consists in augmenting the kernel matrix size when new data become available. A shrink to the original size is proposed when its performance falls below a certain tolerance limit. Then, there are taking into account the last $n$ data. The performance of the first growing phase of the algorithm should be checked after the first iteration; simulating its computational efficiency with random data and establishing a maximum size. Besides of this, overfitting issues should be considered in order to shrink the kernel matrix.

The *worm* windows alternative should offer a major stability in their predictions as consequence of always considering a number of data equal or greater than sliding windows. On the other hand, the sliding alternative requires a minor computational efforts and will take a major proportion of new data. Thus, depending on the nature of the database, its variability, and the targets of the analysis we could choose one of these two options for the on-line learning.

## 3 Numerical results

To validate the on-line MKr approaches introduced in this work, a series of analytical benchmarks have been used, along with a structural design test case.

### 3.1 Analytical test cases

The on-line MKr methods proposed are first validated on a set of three artificial mixed-variable benchmark functions of 5 continuous and 5 discrete variables adapted from [4]. In all cases, we test 20 updates of 5 elements each time. While sliding window strategy multiplied its RMSE error by six along the learning process, the *worm* window error remained nearly constant.

### 3.2 Structural design instance

A structural design example by a 3D rigid frame is also introduced to illustrate the performance of these on-line MKr methods. The quantity of interest is the total mass of the structure which is characterized by ten design, 5 continuous and 5 discrete, variables. Figure 2 shows a comparison between both windowing strategies introduced in this work by the boxplot of their RMSEs.

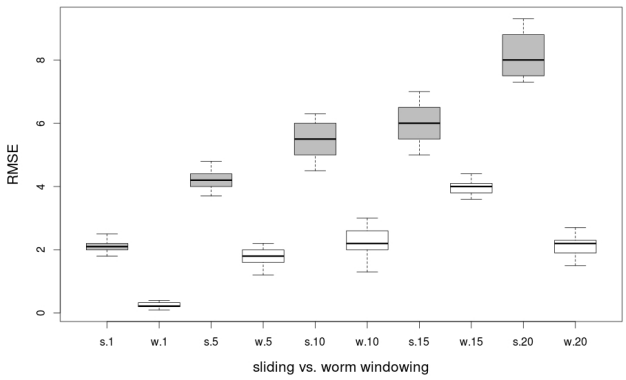



**Fig. 2:** RMSE errors of the windowing strategies. Structural design case-study.

These results support the *worm* window strategy for on-line learning MKr models as a more accurate methodology than sliding windows. In addition, the computational efforts related to *worm* windows can be controlled by a previous tuning phase.

## References

[1] S. C.H. Hoi, Rong Jin, Peilin Zhao, and Tianbao Yang. Online multiple kernel classification. *Machine Learning*, pages 1–27, 2012. In press.

[2] A. Karatzoglou. *Kernel methods software, algorithms and applications*. PhD thesis, 2006.

[3] J. Kivinen, A. Smola, and R. C. Williamson. Online learning with kernels. *IEEE Transactions on Signal Processing*, 52(8):2165–2176, 2004.

[4] T. Liao. Improved ant colony optimization algorithms for continuous and mixed discrete-continuous optimization problems. Technical report, CoDE-IRIDIA Dpt., Université libre de Bruxelles, Belgium, 2011.

[5] H. Lin, D. Chiu, Y. Wu, and A. Chen. Mining frequent itemset from data streams with a time-sensitive sliding window. In *SIAM International Conference on Data Mining 2005*, 2005.

[6] M. Martin. On-line support vector machine regression. In *13th European Conference on Machine Learning - ECML 2002*, pages 282–294, 2002.

[7] S. van Vaerenbergh, J. Vía, and I. Santamaría. A sliding-window kernel RLS algorithm and its application to nonlinear channel identification. In *IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP 2006*, volume V, pages 789–792, 2006.

# Non-parallel semi-supervised classification

Siamak Mehrkanoon
KU Leuven, ESAT-SCD,
Kasteelpark Arenberg 10,
B-3001 Leuven (Heverlee), Belgium.
Siamak.Mehrkanoon@esat.kuleuven.be

Johan A. K. Suykens
KU Leuven, ESAT-SCD,
Kasteelpark Arenberg 10,
B-3001 Leuven (Heverlee), Belgium.
johan.suykens@esat.kuleuven.be

**Abstract**: A non-parallel semi-supervised algorithm based on kernel spectral clustering is formulated. The prior knowledge about the labels is incorporated into the kernel spectral clustering formulation via adding regularization terms. In contrast with the existing multi-plane classifiers such as Multisurface Proximal Support Vector Machine (GEPSVM) and Twin Support Vector Machines (TWSVM) and its least squares version (LSTSVM) we will not use a kernel-generated surface. Instead we apply the kernel trick in the dual. Therefore as opposed to conventional non-parallel classifiers one does not need to formulate two different primal problems for the linear and nonlinear case separately. Experimental results demonstrate the efficiency of the proposed method over existing methods.

**Keywords**: kernel spectral clustering, semi-supervised learning, classification

## 1 Introduction

In the last few years there has been a growing interest in semi-supervised learning in the scientific community. Generally speaking, machine learning can be categorized into two main paradigms, i.e., supervised versus unsupervised learning. Spectral clustering methods, in a completely unsupervised fashion, make use of the eigenspectrum of the Laplacian matrix of the data to divide a dataset into natural groups such that points within the same group are similar and points in different groups are dissimilar to each other. However it has been observed that classical spectral clustering methods suffer from the lack of an underlying model and therefore do not possess an out-of-sample extension naturally. Kernel spectral clustering (KSC) introduced in [1] aims at overcoming these drawbacks. The primal problem of kernel spectral clustering is formulated as a weighted kernel PCA. Recently the authors in [2] have extended the kernel spectral clustering to semi-supervised learning by incorporating the information of labeled data points in the learning process. The concept of having two non-parallel hyperplanes for binary classification was first introduced in [3] where two non-parallel hyperplanes were determined via solving two generalized eigenvalue problem and the method is termed GEPSVM. In this case one obtains two non-parallel hyperplanes where each one is as close as possible to the data points of the one class and as far as possible from the data points of the other class. Some efforts have been made to improve the performance of GEPSVM by providing different formulations such as in [4–6]. It is the purpose here to formulate a non-parallel semi-supervised algorithm based on kernel spectral clustering for which we can directly apply the kernel trick and thus its formulation enjoys the primal and dual properties as in a support vector machines classifier [7].

## 2 Unsupervised and semi-supervised KSC

### 2.1 Primal formulation of binary KSC

The method corresponds to a weighted kernel PCA formulation providing a natural extension to out-of-sample data. Given training data $\{x_i\}_{i=1}^M$, $x_i \in \mathbb{R}^d$ and adopting the model of the following form:

$$e = w^T\varphi(x) + b,$$

the binary kernel spectral clustering in the primal is formulated as follows:

$$\begin{aligned} \min_{w,b,e} \quad & \frac{1}{2}w^Tw - \frac{\gamma}{2}e^TVe \\ \text{subject to} \quad & \Phi w + b1_M = e. \end{aligned}$$

Here $\Phi = [\varphi(x_1), \ldots, \varphi(x_M)]^T$ and a vector of all ones with size $M$ is denoted by $1_M$. $\varphi(\cdot) : \mathbb{R}^d \to \mathbb{R}^h$ is the feature map and $h$ is the dimension of the feature space. $V = \text{diag}(v_1, ..., v_M)$ with $v_i \in \mathbb{R}^+$ is a user defined weighting matrix. It is shown that if $V = D^{-1} = \text{diag}(1/d_1, ..., 1/d_M)$ where $d_i = \sum_{j=1}^M K(x_i, x_j)$ is the degree of the $i-$th data point, the dual problem is related to the random walk algorithm for spectral clustering.

### 2.2 Primal formulation of semi-supervised KSC

KSC is an unsupervised algorithm, by nature, but it has shown its ability to also deal with both labeled and unlabeled data at the same time by incorporating the information of the labeled data into the objective function. Consider training data points $\{x_1, ..., x_N, x_{N+1}, .., x_M\}$ where $\{x_i\}_{i=1}^M \in \mathbb{R}^d$. The first $N$ data points do not have labels whereas the last $N_L = M - N$ points have been labeled with $\{y_{N+1}, .., y_M\}$ in a binary fashion. The information of the labeled samples are incorporated to the binary kernel spectral clustering by means of a regularization term which aims at minimizing the squared distance between the projections of the labeled samples and their corresponding labels [1]:

$$\min_{w,b,e} \quad \frac{1}{2}w^T w - \frac{\gamma}{2}e^T V e + \frac{\rho}{2}\sum_{m=N+1}^{M}(e_m - y_m)^2$$
$$\text{subject to} \quad \Phi w + b1_M = e,$$

where $V = D^{-1}$ is defined as previously. Using the KKT optimality conditions one can show that the solution in the dual can be obtained by solving a linear system of equations [2]. The model selection is done by using an affine combination (where the weight coefficients are positive) of a Fisher criterion and classification accuracy for labeled data.

## 3 Non-parallel semi-KSC

Suppose the training data set $\mathcal{X}$ consists of $M$ data points and is defined as follows:

$$\mathcal{X} = \{\underbrace{x_1, ..., x_N}_{Unlabeled\ (\mathcal{X}_U)}, \underbrace{x_{N+1}, .., x_{N+\ell_1}}_{Labeled\ with\ (+1)\ (\mathcal{X}_{L_1})}, \underbrace{x_{N+\ell_1+1}, .., x_{N+\ell_1+\ell_2}}_{Labeled\ with\ (-1)\ (\mathcal{X}_{L_2})}\},$$

where $\{x_i\}_{i=1}^M \in \mathbb{R}^d$. The target values are denoted by set $\mathcal{Y}$ which consists of binary labels:

$$\mathcal{Y} = \{\underbrace{+1, \ldots, +1}_{y^1}, \underbrace{-1, \ldots, -1}_{y^2}\}.$$

We seek two non-parallel hyperplanes:

$$f_1(x) = w_1^T\varphi(x) + b_1 = 0, \quad f_2(x) = w_2^T\varphi(x) + b_2 = 0,$$

where each one is as close as possible to the points of its own class and as far as possible from the data of the other class. We formulate a non-parallel semi-supervised KSC, in the primal, as the following two optimization problems [8]:

$$\begin{aligned}\min_{w_1,b_1,e,\eta,\xi} \quad & \frac{1}{2}w_1^T w_1 + \frac{\gamma_1}{2}\eta^T\eta + \frac{\gamma_2}{2}\xi^T\xi - \frac{\gamma_3}{2}e^T D^{-1}e \\ \text{subject to} \quad & w_1^T\varphi(x_i) + b_1 = \eta_i,\ \forall x_i \in I, \\ & y_i^2\left[w_1^T\varphi(x_i) + b_1\right] + \xi_i = 1,\ \forall x_i \in II, \\ & w_1^T\varphi(x_i) + b_1 = e_i,\ \forall x_i \in \mathcal{X}, \end{aligned} \tag{1}$$

where $\gamma_1, \gamma_2$ and $\gamma_3 \in \mathbb{R}^+, b_1 \in \mathbb{R}$, $\eta \in \mathbb{R}^{N_{L1}}$, $\xi \in \mathbb{R}^{N_{L2}}$, $e \in \mathbb{R}^M$, $w_1 \in \mathbb{R}^h$. $\varphi(\cdot) : \mathbb{R}^d \to \mathbb{R}^h$ is the feature map and $h$ is the dimension of the feature space.

$$\begin{aligned}\min_{w_2,b_2,e,\rho,\nu} \quad & \frac{1}{2}w_2^T w_2 + \frac{\gamma_4}{2}\rho^T\rho + \frac{\gamma_5}{2}\nu^T\nu - \frac{\gamma_6}{2}e^T D^{-1}e \\ \text{subject to} \quad & w_2^T\varphi(x_i) + b_2 = \rho_i,\ \forall x_i \in II, \\ & y_i^1\left[w_2^T\varphi(x_i) + b_2\right] + \nu_i = 1,\ \forall x_i \in I, \\ & w_2^T\varphi(x_i) + b_2 = e_i,\ \forall x_i \in \mathcal{X}, \end{aligned} \tag{2}$$

where $\gamma_4, \gamma_5$ and $\gamma_6 \in \mathbb{R}^+$. $b_2 \in \mathbb{R}$, $\rho \in \mathbb{R}^{N_{L2}}$, $\nu \in \mathbb{R}^{N_{L1}}$, $e \in \mathbb{R}^M$, $w_2 \in \mathbb{R}^h$. $\varphi(\cdot)$ is defined as previously.

The performance of the Semi-KSC [2] and the proposed method in this paper when a linear kernel is used are shown in Fig. 1. The result of the proposed method (NP-Semi-KSC) is compared with that of Semi-KSC, Laplacian SVM (LapSVM) [9] and its recent version LapSVMp [10] recorded in [2] over some benchmark datasets for semi-supervised learning. When few labeled data points are available the proposed method shows a comparable result with respect to other methods. But as the number of labeled data points increases NP-Semi-KSC outperforms in most cases the other methods.

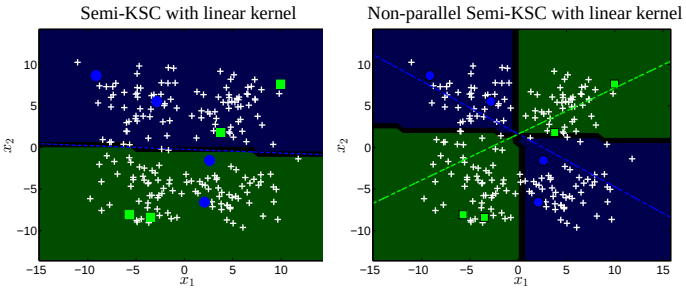



**Fig. 1:** Toy Problem-Four Gaussians with some overlap. The training and validation parts consist of $N_{tr} = 100$ and $N_{val} = 100$ unlabeled data points respectively. The labeled data points of two classes are depicted by the blue squares and green circles. (a): Result of semi-supervised kernel spectral clustering when linear kernel is used. The separating hyperplane is shown by blue dashed line. (b): Result of the proposed non-parallel semi-supervised KSC when linear kernel is used. Two non-parallel hyperplanes are depicted by blue and green dashed lines.

**Tab. 1:** Average misclassification test error $\times 100\%$. The calculation of the test error is done by evaluating the methods on the full data sets. Two cases for the labeled data (LD) size are considered (i.e. # labeled data points=10 and 100).

| # of LD | Method | g241c | g241d | BCI |
|---|---|---|---|---|
| 10 | LapSVM | $0.48 \pm 0.02$ | $0.42 \pm 0.03$ | $0.48 \pm 0.03$ |
| | LapSVMp | $0.49 \pm 0.01$ | $0.43 \pm 0.03$ | $0.48 \pm 0.02$ |
| | Semi-KSC | $\mathbf{0.42 \pm 0.03}$ | $0.43 \pm 0.04$ | $\mathbf{0.46 \pm 0.03}$ |
| | NP-Semi-KSC | $0.44 \pm 0.03$ | $\mathbf{0.41 \pm 0.02}$ | $0.47 \pm 0.03$ |
| 100 | LapSVM | $0.40 \pm 0.06$ | $0.31 \pm 0.03$ | $0.37 \pm 0.04$ |
| | LapSVMp | $0.36 \pm 0.07$ | $0.31 \pm 0.02$ | $0.32 \pm 0.02$ |
| | Semi-KSC | $0.29 \pm 0.05$ | $0.28 \pm 0.05$ | $0.28 \pm 0.02$ |
| | NP-Semi-KSC | $\mathbf{0.23 \pm 0.01}$ | $\mathbf{0.26 \pm 0.02}$ | $\mathbf{0.26 \pm 0.01}$ |


## Acknowledgments

This work was supported by: ● Research Council KUL: GOA/10/09 MaNet, PFV/10/002 (OPTEC), several PhD/postdoc & fellow grants ● Flemish Government: ○ IOF: IOF/KP/SCORES4CHEM;○ FWO: PhD/postdoc grants, projects: G.0320.08 (convex MPC), G.0558.08 (Robust MHE), G.0557.08 (Glycemia2), G.0588.09 (Brain-machine), G.0377.09 (Mechatronics MPC); G.0377.12 (Structured systems) research community (WOG: MLDM); ○ IWT: PhD Grants, projects: Eureka-Flite+, SBO LeCoPro, SBO Climaqs, SBO POM, O&O-Dsquare ● Belgian Federal Science Policy Office: IUAP P7/ (DYSCO, Dynamical systems, control and optimization, 2012-2017) ● IBBT ● EU: ERNSI, FP7-EMBOCON (ICT-248940), FP7-SADCO (MC ITN-264735), ERC ST HIGHWIND (259 166), ERC AdG A-DATADRIVE-B ● COST: Action ICO806: IntelliCIS ● Contract Research: AMINAL ● Other: ACCM. Johan Suykens is a professor at the KU Leuven, Belgium.


## References


[1] C. Alzate and J. A. K. Suykens, "Multiway spectral clustering with out-of-sample extensions through weighted kernel PCA," *IEEE Trans- actions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 2, pp. 335-347, 2010.

[2] C. Alzate and J. A. K. Suykens, "A Semi-Supervised Formulation to Binary Kernel Spectral Clustering," *In Proc. of the 2012 IEEE World Congress on Computational Intelligence*, Brisbane, 1992-1999, 2012.

[3] O.L. Mangasarian and E.W. Wild, "Multisurface Proximal Support Vector Machine Classification via Generalized Eigenvalues," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 1, pp. 69-74, 2006.

[4] Jayadeva, R. Khemchandani and S. Chandra, "Twin Support Vector Machines for Pattern Classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 5, pp.905-910, 2007.

[5] Y.H. Shao, C.H. Zhang, X.B Wang, and N.Y. Deng, "Improvements on Twin Support Vector Machines," *IEEE Transactions on Neural Networks*, vol. 22, no. 6, pp. 962-968, 2011.

[6] M.A. Kumar, M. Gopal, "Least squares twin support vector machines for pattern classification," *Expert Systems with Applications*, 36, 7535-7543, 2009.

[7] J.A.K. Suykens, C. Alzate, K. Pelckmans, "Primal and dual model representations in kernel-based learning", *Statistics Surveys*, vol. 4, pp. 148-183, 2010.

[8] S. Mehrkanoon, J.A.K. Suykens, "Non-parallel semi-supervised classification based on kernel spectral clustering", accepted for publication in IJCNN 2013.

[9] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *J. Mach. Learn. Res.*, vol. 7, pp. 2399-2434, 2006.

[10] S. Melacci and M. Belkin, "Laplacian support vector machines trained in the primal," *Journal of Machine Learning Research*, 12, 1149-1184, 2011.

# Visualisation of neural networks for model reduction

Tamás Kenesei
University of Pannonia,
Department of Process Engineering,
P.O. Box 158, H-8201
tamas.kenesei@gmail.com

János Abonyi
University of Pannonia,
Department of Process Engineering,
P.O. Box 158, H-8201
janos@abonyilab.com

**Abstract**: Neural networks are difficult to interpret. Extraction and visualisation of fuzzy if-then rules from neural networks can support the analysis of these black-box models. We overview existing rule-extraction and model reduction methods and propose a technique based on structural equality between sigmoid transfer function based neural networks and fuzzy systems. Visualisation is based on similarity of membership functions represented by sigmoid transfer functions. Orthogonal least squares (OLS) and merging of fuzzy sets are used for model pruning. An illustrative example is shown to demonstrate the effectiveness of the method. This simple example shows that visualization and similarity based ranking of neurons help the user to find redundant parts of the model, while OLS based ranking is useful to decide which elements can be removed without significant performance loss.

**Keywords**: model reduction, model visualisation, regularisation

## 1 Introduction

Neural networks (NNs) are efficient in nonlinear regression and classification. Main disadvantage of NNs is that they are not interpretable. Other problem is how *a–priori* knowledge can be utilised and integrated into this black-box modeling approach, and how a human expert can validate the identified NNs. To overcome these problems, following strategies can be used:

1. **Transformation of NN.** Convert NN into a more interpretable form. A good approach is to extract fuzzy if-then rules from NN [1, 2].
2. **Model reduction.** Overcome complexity problems with 'importance' determination of hidden neurons and weights, remove the insignificant ones, and/or merge the similar ones. Regularised training can also be considered as this approach.
3. **Visualisation of neural network.** Generate a two dimensional map of the neurons and utilise human expert to evaluate the structure.

We combine these methods into an effective tool for iterative and interactive modelling tool. The neural network is transformed into a fuzzy rule base. The fuzzy sets at the antecedent part of the rules are merged based on a similarity measure. Similarities of the rules are also calculated. These values are used as distance measure for the visualisation of the rulebase. This technique can be used as unsupervised approach for model reduction. As performance based model reduction orthogonal least squares technique is applied.

## 2 Model Transformation

Logistic activation function based NN's are structurally identical to a special type of fuzzy rule based model - called fuzzy additive system (FAS) [2]. In order to keep the equality relationship between the NN and a corresponding fuzzy rule-based system, a new logical operator has been presented. Using this $*$ operator, fuzzy rules extracted from the trained NN can be represented as follows:

$$\mathbf{R}_j: \textbf{If } x_1 \textit{ is } A_j^1 \ * \ldots * \ x_n \textit{ is } A_j^n \textbf{ then } y \ = b_j \quad (1)$$

where $*$ represents the interactive-or or $i$-or: $a * b = (ab)\backslash((1-a)(1-b)+ab)$ operator is used to decompose the multivariate logistic function into *univariate* membership functions of the rule antecedents.

## 3 Visualisation and model reduction

The antecedent part of the rules can be analysed based on the similarity of the membership functions, $S_{j,l}^i$

$$S_{j,l}^i = \frac{A_j^i \cap A_l^i}{A_j^i \cup A_l^i} \quad (2)$$

With this measure, pairwise similarities of hidden neurons in the range of $[0, 1]$ can be obtained:

$$S_{j,l} = \prod_i S_{j,l}^i, \ i = 1, \ldots, n. \quad (3)$$

We apply multidimensional scaling to generate a map of neurons by preserving the distances $d_{j,l} = 1 - S_{j,l}$

among the neurons. These maps can be effectively used to extract the hidden structure of the network and control the merging of similar neurons (membership functions). Among the most similar pairs unnecessary rules (and neurons) with low error reduction ratio should be removed. Since the consequent parameters of the rules are linear in parameters orthogonal least squares techniques can evaluate the individual contribution of the rules [3].

## 4 Application example

Neural networks are widely used in modelling of dynamical systems, mostly in the NARX model form, $y(k+1) = f\left(y(k), \ldots, y(k-n_y), u(k), \ldots, u(k-n_d)\right)$, where $y$ represents the output, $u$ the input of the system, $k$ stands for the discrete time instant and $n_y$ and $n_u$ are the input and output orders of the dynamical system. In these problems the identification of the proper model structure $f(.)$ is extremely difficult since the performance of the model is sensitive to overparametrisation. The proposed visualisation and reduction techniques are applied to model a continuous stirred tank reactor [4], where the actual output (the pH) $y(k+1)$, depends on the state of the reactor (the previous pH value $y(k)$ and the NaOH feed $u(k)$ at the $k$th sample time:

$$y(k+1) = f\left(y\left(k\right), u\left(k\right)\right) \tag{4}$$

Parameters of the neural network were identified by the back-propagation algorithm based on a uniformly distributed training data where $F_{NaOH}$ is in the range of 515-525 l/min. Our experiences show that 7 neurons are sufficient in the hidden layer of the NN. Figure 1 shows the membership functions extracted from this neural network. Analysis of the mapping of the neurons (see Fig. 2.) shows that it is possible to remove one neuron since the 2nd and the 7th neurons are closer to each other. OLS based ranking indicates the 2nd rule more important. Therefore the 7th neuron can be removed model without a significant decrease in modelling performance. Result indicates outstanding performance of the reduced model even by free run simulation (The mean square error is $3.5 \cdot 10^{-3}$ for 7 neurons, and $3.824 \cdot 10^{-3}$ for 6 neurons). This simple example shows that visualization and similarity based ranking of neurons help the user to find redundant parts of the model, while OLS based ranking is useful to decide which elements of these paris could be removed without significant loss of performance.

## 5 Conclusion

Visualisation of hidden structure of neural networks can support iterative and interactive model reduction. Similarity of neurons is calculated based on the similarity of antecedent fuzzy sets extracted from the neural network model.

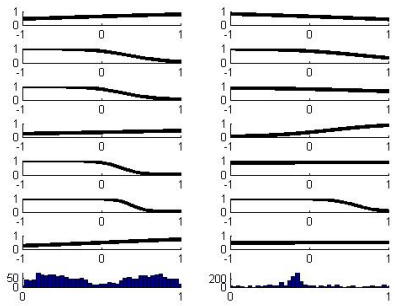


**Fig. 1:** Decomposed univariate membership functions

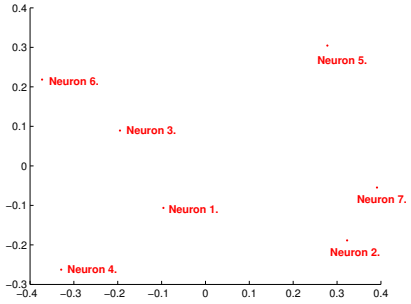



**Fig. 2:** Distances between neurons mapped into two dimensions with MDS. Please note, the axes does not have physical meaning, the plot represenst the distances, $d_{j,l} = 1 - S_{j,l}$ of the neurons.

Orthogonal least squares technique is used to evaluate the contribution of the individual rules to support the user to decide which neuron could be removed from the model without significant decrease of prediction performance.

**Acknowledgments**
This publication/research has been supported by the projects TAMOP-4.2.2.A-11/1/KONV-2012-0071 and TAMOP-4.2.2/B-10/1-2010-0025.

## References

[1] Huynh, T. Q. & Reggia, J. Guiding hidden layer representations for improved rule extraction from neural networks. IEEE Trans. on Neural Networks, 22(2), 264-275., 2011

[2] Castro, J. L., Mantas, C. J., & Benitez, J. M. Interpretation of artificial neural networks by means of fuzzy rules. IEEE Trans. on Neural Networks, 13(1), 101-116. , 2002.

[3] O. Nelles Nonlinear system identification *Springer-Verlag*, 2001.

[4] J. Abonyi Fuzzy Model Identification for Control *Birkhauser*, 2002.

# Convergence analysis of stochastic gradient descent on strongly convex objective functions

Cheng Tang
Department of Computer Science,
The George Washington University
tangch@gwu.edu

Claire Monteleoni
Department of Computer Science,
The George Washington University
cmontel@gwu.edu

**Abstract**: Recently, [3] posed an open problem on whether the Stochastic Gradient Descent (SGD) algorithm, without averaging, could achieve the optimal $O(\frac{1}{t})$ error convergence rate on strongly convex (possibly non-smooth) functions. Our work gives an affirmative answer to this question for a subclass of functions. In high dimension, challenges arise in uniting the analyses of different optimality conditions for general strongly convex functions. We provide a subclassification scheme for these functions, using a refined definition of strong convexity and a relaxed definition of strong smoothness. We show how each definition captures the class of strongly convex functions and how we can describe the lower and upper bound on the norm of the subgradients of the function, which characterizes the difficulty of optimizing it. Our approach provides a roadmap for future work.

**Keywords**: stochastic optimization, gradient descent, strong convexity

## Introduction

Stochastic Gradient Descent (SGD) is a simple optimization method for solving a convex program. It is the prototypical algorithm to solve online and large scale batch machine learning problems due to its simplicity and efficiency. Strongly convex functions are often adopted in learning problems to formalize a regularized objective, such as in SVM [4]. While most of the earlier work assumes twice differentiability of the function $F$, our setting follows a more general framework studied in [1, 2]: we consider the convergence rate of the last iterate returned by SGD on $F$.

Our assumptions are: **1.** $F$ is $\lambda$-strongly convex and has a bounded convex domain $W$. **2.** We have oracle access to an unbiased estimator of a subgradient of $F$ at any point $w$, denoted $\hat{g}, i.e., E[\hat{g}] \in \partial F(w)$, with $\|\hat{g}\|_2 \leq G$. That is, the norm of the subgradients are all bounded by some number $G$.

We apply SGD with the update rule: $w^{t+1} = \Pi_W(w_t - \eta_t \hat{g}_t)$, where $\Pi_W$ is the projection operator on $W$. We set the learning rate $\eta_t$ to be $\frac{c}{\lambda t}$, with $c \geq 2$.

Our goal is to determine whether SGD without averaging has optimal convergence rate on strongly convex functions in terms of the expected error, $E[F(w_t) - F(w_{opt})]$. Since SGD is often used to solve problems with large datasets, this memory-free version of SGD will have an obvious advantage and should be adopted if it can be shown to simultaneously match the optimal performance of its computationally more demanding counterparts.

## Results

We introduce a scheme to subclassify the strongly convex functions according to their regularity conditions at the optimum, $w_{opt}$. Previously, [1] points out that if $F$ is strongly smooth, or very "flat", at $w_{opt}$, SGD has an optimal convergence rate. On the other hand, [3] observes that if $F$ is very "pointy" at $w_{opt}$, SGD should have a fast descent and conjecture that in this case SGD should also achieve the optimal rate. Our approach generalizes these observations: regularity condition at $w_{opt}$ determines the "shape" of $F$. Strong convexity at $w_{opt}$ implies that $F$ is more "pointy" around the optimum while strong smoothness implies $F$ is "flatter". Thus, convexity and smoothness characterize two opposite properties of $F$. Due to our restriction that $F$ is strongly convex, it can not become any "flatter" than a quadratic function at $w_{opt}$ while at the extreme end of strong convexity, $F$ can be linear at $w_{opt}$ in each direction. Based on this insight, we first relax the notion of local strong smoothness to local weak smoothness.

**Definition 1.** *Let $\nu \in [0, 1]$. $F$ is $(\nu, L_\nu)$-*__local weakly smooth__*, if $\exists L_\nu > 0$ s.t. $\forall w \in W$ and $\forall g(w) \in \partial F(w)$ and $\forall g(w_{opt}) \in \partial F(w_{opt})$, we have $\|g(w) - g(w_{opt})\| \leq L_\nu \|w - w_{opt}\|^\nu$*

**Definition 2.** *$F \in H^\nu$, if $F$ is $(\nu, L_\nu)$-*__local weakly smooth__*.*

The nested family of local weakly smooth functions is shown in **Fig. 1**. Since all strongly convex functions

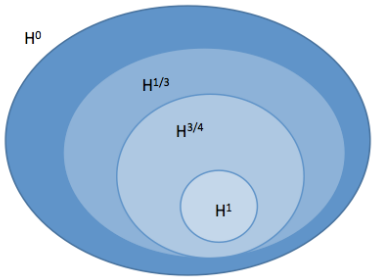



**Fig. 1:** subclassification by local weak smoothness

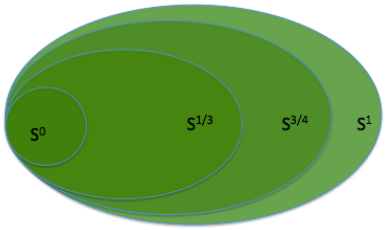



**Fig. 2:** subclassification by refined local strong convexity

with finite subgradients form a subset of $H^0$, each $F$ in our framework is contained in multiple $H^\nu$. But for each $F$, $\exists \nu_{\max} > 0$ such that $H^{\nu_{\max}}$ is the largest $H^\nu$ containing $F$, which characterizes the local smoothness, or "flatness" of $F$. $\nu_{\max}$ in $[0, 1]$ provides an upper bound on $g(w)$ as a polynomial function of $\|w - w_{opt}\|$. Similarly, we refine the notion of strong convexity to subclassify local strongly convex functions, which was first defined by [5] in their work with a different goal.

**Definition 3.** *Let $\mu \in [0, 1]$. $F$ is $(\mu, \lambda_\mu)$-*__local strongly convex__*, if $\exists \lambda_\mu > 0$ s.t. $\forall w \in W$ and $\forall g(w) \in \partial F(w)$ and $\forall g(w_{opt}) \in \partial F(w_{opt})$, $\|g(w) - g(w_{opt})\| \geq \lambda_\mu \|w - w_{opt}\|^\mu$*

**Definition 4.** *$F \in S^\mu$, if $F$ is $(\mu, \lambda_\mu)$-*__local strongly convex__*.*

The strongly convex functions form another nested family as shown in **Fig. 2**. $\mu$ characterizes the degree of local strong convexity, or "pointiness" of the function at $w_{opt}$ in a similar way. We emphasize that even though we use the parameters $\nu$ and $\mu$ in our analysis, we do not assume the SGD algorithm knows these parameters. Thus, our analysis does not impose additional assumptions on the strongly convex functions being studied.

Our next result confirms our intuition that if $F$ is non-differentiable at $w_{opt}$ from every direction, or $F \in S^0$, then the SGD attains the optimal convergence rate due to fast descents.

**Theorem 1.** *Suppose $\exists L_0 > 0$ s.t. $\forall w_t \neq w_{opt}$, $F(w_t) - F(w_{opt}) \geq L_0 \|w_t - w_{opt}\|$ . Consider SGD with step sizes $\eta_t = \frac{c}{\lambda t}$, with $c \geq 2$. For any $T > 1$, it holds that*

$$E[F(w_t) - F(w_{opt})] \leq G\sqrt{M}\frac{c}{\lambda t}$$

*where $M = \max\left\{\frac{G^4}{4L_0^2}, 16(G + L_0)^2\right\}$*

## Future work

Our subclassification scheme generalizes the idea that smoothness at $w_{opt}$ provides a bound on the variance of the error rate. Previous results require local strong smoothness, which corresponds to $F \in S1 \cap H^0$. For future work, we propose only assuming local weak smoothness, or $F \in S^1 \cap \{\cup_{\nu>0} H^\nu\}$, which subsumes local strong smoothness and varies continuously according the parameter $\nu$. Based on this scheme, future work on proving optimal convergence on the subclass of local weakly smooth functions might be developed.

**Acknowledgements**: We thank anonymous reviewers of earlier versions, for their comments used in revision.

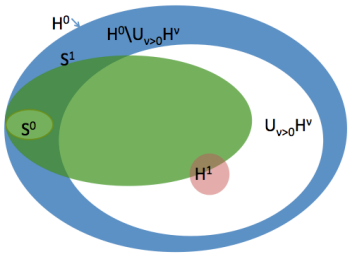



**Fig. 3:** A visualization of the superposition of the two subclassification schemes

In **Fig. 3**: The green ellipse represents the class of all the strongly convex functions. We proved the optimal convergence of SGD on the subset labeled $S^0$. Previous analysis showed an optimal convergence on the intersection of the green ellipse and the red circle. For future work, our proposed subclassification scheme can be used towards extending the result to all of the light green region.

## References

[1] Alexander Rakhlin, Ohad Shamir and Karthik Sridharan *Making Gradient Descent Optimal for Strongly Convex Stochastic Optimization CoRR*, abs/1109.5647, 2011.

[2] Elad Hazan and Satyen Kale *Beyond the regret minimization barrier: an optimal algorithm for stochastic strongly-convex optimization.* Journal of Machine Learning Research - Proceedings Track, 19, 2011.

[3] Ohad Shamir. *Is averaging needed for Strongly Convex Stochastic Gradient Descent?*. COLT open problem, 2012.

[4] Shai Shalev-Shwartz, Yoram Singer, and Nathan Srebro. *Pegasos: Primal Estimated sub-GrAdient SOlver for SVM.* ICML, 2007.

[5] Aaditya Ramdas and Aarti Singh. *Optimal Stochastic Convex Optimization Through The Lens Of Active Learning. CoRR*, abs/1207.3012, 2012.

# List of participants

Prof. Janos Abonyi
University of Pannonia
POB. 158
8200 Veszprem - Hungary
janos@abonyilab.com

Masoud Ahookhosh
University of Vienna
Nordbergstrasse 15
1090 Vienna - Austria
masoud.ahookhosh@univie.ac.at

Reema Al-Aifari
Vrije Universiteit Brussel
Pleinlaan 2
Vakgroep Wiskunde
1050 Brussels - Belgium
reema@alaifari.com

Dr. Carlos Alzate
IBM Research
Smarter Cities Technology Centre
Damastown Industrial Estate
Dublin 15 Mulhuddart - Ireland
carlos.alzate@ie.ibm.com

Prof. Andreas Argyriou
Ecole Centrale Paris
2, Avenue Sully Prudhomme
92290 Chatenay-Malabry - France
andreas.argyriou@ecp.fr

Prof. Francis Bach
Ecole Normale Superieure Paris
23, Avenue d'Italie
CS 81321
75214 Paris Cedex 13 - France
francis.bach@ens.fr

Dr. Alper Bilge
Anadolu University
2 Eylul
Kampusu M.M.F. Bilgisayar
26470 Eskisehir - Turkey
abilge@anadolu.edu.tr

Prof. Stephen Boyd
Stanford University
Packard 264
CA 94305 Stanford - United States
boyd@stanford.edu

Dr. Kris De Brabanter
KU Leuven, ESAT-SCD
Kasteelpark Arenberg 10
3001 Leuven-Heverlee - Belgium
kris.debrabanter@esat.kuleuven.be

Prof. Christine De Mol
Universite libre de Bruxelles
Department of Mathematics
Campus Plaine CPI 217
Boulevard du Triomphe
1050 Brussels - Belgium
demol@ulb.ac.be

Dr. Francesco Dinuzzo
MPI for Intelligent Systems
Spemannstrasse 38
72076 Tuebingen - Germany
fdinuzzo@tuebingen.mpg.de

Prof. Jose Dorronsoro
ADIC
EPS-UAM
C Tomas y Valiente 11
28049 Madrid - Spain
jose.dorronsoro@uam.es

Dr. Orla Doyle
King's College London
Centre for Neuroimaging Sciences
Box 089, Institute of Psychiatry
De Crespigny Park
London SE58AF - United Kingdom
orla.doyle@kcl.ac.uk

Prof. Pierre Dupont
Catholic University of Louvain - UCL
Machine Learning Group - ICTEAM
Place Sainte-Barbe 2
1348 Louvain-la-Neuve - Belgium
Pierre.Dupont@uclouvain.be

Dr. Tillmann Falck
KU Leuven, ESAT-SCD
Kasteelpark Arenberg 10
3001 Leuven-Heverlee - Belgium
tillmann.falck@esat.kuleuven.be

Basura Fernando
KU Leuven, ESAT-PSI/VISICS
Kasteelpark Arenberg 10
3001 Leuven-Heverlee - Belgium
basura.fernando@esat.kuleuven.be

Dr. Alessandro Ghio
University of Genoa
Via Opera Pia 11A
16145 Genoa - Italy
alessandro.ghio@unige.it

Amir Ghodrati
KU Leuven, ESAT-PSI
Kasteelpark Arenberg 10
3001 Leuven-Heverlee - Belgium
amir.ghodrati@esat.kuleuven.be

Dr. Nicolas Gillis
Universite catholique de Louvain - UCL
Avenue G. Lemaitre, 4 - bte L4.05.01
1348 Louvain-la-Neuve - Belgium
nicolas.gillis@uclouvain.be

Prof. Francois Glineur
Universite catholique de Louvain - UCL
CORE
Voie du Roman Pays 34 bte L1.03.01
1348 Louvain-la-Neuve - Belgium
francois.glineur@uclouvain.be

Paul Grigas
Massachusetts Institute of Technology
77 Massachusetts Ave.
E40-149
02139 Cambridge - United States
pgrigas@mit.edu

Emmanuel Herbert
Orange Labs
173, rue lafayette
75010 Paris - France
emmanuel.herbert@orange.com

Dr. Manuel Herrera
Universite libre de Bruxelles
Avenue F. Roosevelt, 50 (CP 194/2)
1050 Bruxelles - Belgium
mherrera@ulb.ac.be

Pablo Hess
TRYCON GCM AG
Rheinstrasse 13
60325 Frankfurt - Germany
hess@trycongcm.com

Dr. Xiaolin Huang
KU Leuven, ESAT-SCD
Kasteelpark Arenberg 10
3001 Leuven-Heverlee - Belgium
xiaolin.huang@esat.kuleuven.be

Dr. Mariya Ishteva
Vrije Universiteit Brussel
Pleinlaan 2, Building K
1050 Brussels - Belgium
mariya.ishteva@vub.ac.be

Prof. Martin Jaggi
Ecole Polytechnique, Paris
37, Boulevard Jourdan
75014 Paris - France
m.jaggi@gmail.com

Prof. Kurt Jetter
Universitaet Hohenheim
Obere Schneeburgstr. 38
79111 Freiburg - Germany
Kurt.Jetter@uni-hohenheim.de

Vilen Jumutc
KU Leuven, ESAT-SCD
Kasteelpark Arenberg 10
3001 Leuven-Heverlee - Belgium
Vilen.Jumutc@esat.kuleuven.be

Dr. Cihan Kaleli
Anadolu University
2 Eylul
Kampusu M.M.F. Bilgisayar
26470 Eskisehir - Turkey
ckaleli@anadolu.edu.tr

Dr. Atsushi Kawamoto
Toyota Central R&D Labs., Inc.
Nagakute
480-1192 Aichi - Japan
atskwmt@mosk.tytlabs.co.jp

Dr. Hyon-Jung Kim
Aalto University
Otakaari 5 A
02150 Espoo - Finland
hyon-jung.kim-ollila@aalto.fi

Parisa Kordjamshidi
KU Leuven, Dep. Computer Science
Celestijnenlaan 200A
3001 Leuven-Heverlee - Belgium
Parisa.Kordjamshidi@cs.kuleuven.be

Anna Krause
Gottfried Wilhelm Leibniz Universitaet
Hannover
Institut fuer Mikroelektronische Systeme
Appelstrasse 4
30167 Hannover - Germany
anna.krause@ims.uni-hannover.de

Prof. James Kwok
Hong Kong University of Science
and Technology
Clear Water Bay
Hong Kong - China
jamesk@cse.ust.hk

Prof. Gert Lanckriet
University of California, San Diego
9500 Gilman Drive
92093-0407 San Diego - United States
glanckriet@eng.ucsd.edu

Rocco Langone
KU Leuven, ESAT-SCD
Kasteelpark Arenberg 10
3001 Leuven-Heverlee - Belgium
Rocco.Langone@esat.kuleuven.be

Augustin Lefevre
Universite catholique de Louvain - UCL
Avenue Georges Lemaitre 4
1348 Louvain-la-Neuve - Belgium
augustin.lefevre@uclouvain.be

Prof. Ignace Loris
Universite libre de Bruxelles
Departement de Mathematique
Boulevard du Triomphe
1050 Bruxelles - Belgium
igloris@ulb.ac.be

Raghvendra Mall
KU Leuven, ESAT-SCD
Kasteelpark Arenberg 10
3001 Leuven-Heverlee - Belgium
raghvendra.mall@esat.kuleuven.be

Prof. Ivan Markovsky
Vrije Universiteit Brussel
Pleinlaan 2, Building K
1050 Brussels - Belgium
imarkovs@vub.ac.be

Dr. Andre Marquand
King's College London
Institute of Psychiatry, Box P089
De Crespigny Park
London SE173RT - United Kingdom
andre.marquand@kcl.ac.uk

Vladimir Matic
KU Leuven, ESAT-SCD
Kasteelpark Arenberg 10
3001 Leuven-Heverlee - Belgium
vladimir.matic@esat.kuleuven.be

Siamak Mehrkanoon
KU Leuven, ESAT-SCD
Kasteelpark Arenberg 10
3001 Leuven-Heverlee - Belgium
Siamak.Mehrkanoon@esat.kuleuven.be

Prof. Marie-Francine Moens
KU Leuven, Dep. Computer Science
Celestijnenlaan 200A
3001 Leuven-Heverlee - Belgium
Sien.Moens@cs.kuleuven.be

Prof. Claire Monteleoni
George Washington University
Computer Science Department
801 22nd St. NW, Suite 703
20052 Washington, DC - United States
cmontel@gwu.edu

Vahid Nassiri
Vrije Universiteit Brussel
15, rue de la Jonchaie
1040 Brussels - Belgium
vnassiri@vub.ac.be

Dr. Valeriya Naumova
Johann Radon Institute for
Computational and Applied Mathematics
Altenbergerstrasse 69
4040 Linz - Austria
valeriya.naumova@oeaw.ac.at

Prof. Yurii Nesterov
Catholic University of Louvain - UCL
CORE, 34 voie du Roman Pays
1348 Louvain-la-Neuve - Belgium
yurii.nesterov@uclouvain.be

Dr. Esa Ollila
Aalto University
Otakaari 5 A
02150 Espoo - Finland
esa.ollila@aalto.fi

Francesco Orsini
KU Leuven, Dep. Computer Science
Celestijnenlaan 200A
3001 Leuven-Heverlee - Belgium
Francesco.Orsini@cs.kuleuven.be

Dr. Marco Pedersoli
KU Leuven, ESAT-PSI/VISICS
Kasteelpark Arenberg 10
3001 Leuven-Heverlee - Belgium
marco.pedersoli@esat.kuleuven.be

Prof. Kristiaan Pelckmans
Uppsala University
Department of Information Technology
Box 337
751 05 Uppsala - Sweden
kristiaan.pelckmans@it.uu.se

Prof. Christophe Phillips
Univerity of Liege
Cyclotron Research Centre, B30
Sart Tilman
4000 Liege - Belgium
c.phillips@ulg.ac.be

Prof. Massimiliano Pontil
University College London
Malet Place
London WC1E 6BT - United Kingdom
m.pontil@cs.ucl.ac.uk

Thomas Provoost
KU Leuven, Dep. Computer Science
Celestijnenlaan 200A
3001 Leuven-Heverlee - Belgium
Thomas.Provoost@cs.kuleuven.be

Jose Gervasio Puertas
KU Leuven, ESAT-SCD
Kasteelpark Arenberg 10
3001 Leuven-Heverlee - Belgium
gpuertas@esat.kuleuven.be

Prof. Jan Ramon
KU Leuven, Dep. Computer Science
Celestijnenlaan 200A
3001 Leuven-Heverlee - Belgium
Jan.Ramon@cs.kuleuven.be

Dr. Wojciech Rejchel
Nicolaus Copernicus University
Faculty of Mathematics and
Computer Science
Chopina 12/18
87-100 Torun - Poland
wrejchel@gmail.com

Flamary Remi
Universite de Nice Sophia-Antipolis
Laboratoire Lagrange, UMR 7293
Parc Valrose
06108 Nice - France
remi.flamary@unice.fr

Prof. Justin Romberg
Georgia Institute of Technology
777 Atlantic Drive NW
30332 Atlanta - United States
jrom@ece.gatech.edu

Dr. Maria Rosa
University College London
66a Kellett Road
London SW2 1ED - United Kingdom
m.rosa@ucl.ac.uk

Alessandro Rudi
Istituto Italiano di Tecnologia
Via Morego 30
16163 Genova - Italy
alessandro.rudi@gmail.com

Prof. Tobias Ryden
Lynx Asset Management
Norrmalmstorg 12
Box 7060
10386 Stockholm - Sweden
tobias.ryden@lynxhedge.se

Jose Sanchez
Chalmers University of Technology
Matematiska Vetenskaper
Chalmers tvargata 3
412 96 Gothenburg - Sweden
sanchezj@chalmers.se

Prof. Bernhard Schoelkopf
Max Planck Institute Tuebingen
Spemannstrasse 38
72076 Tuebingen - Germany
bernhard.schoelkopf@tuebingen.mpg.de

Dr. Jessica Schrouff
Cyclotron Research Centre
University of Liege
8, Allee du 6-Aout, B30
4000 Liege - Belgium
jschrouff@ulg.ac.be

Dr. Fermin Segovia-Roman
University of Liege
Centre de recherches du cyclotron
(Bat. B30)
8, Allee du 6 Aout
4000 Liege - Belgium
fsegovia@ulg.ac.be

Dr. Marko Seslija
KU Leuven, ESAT-SCD
Kasteelpark Arenberg 10
3001 Leuven-Heverlee - Belgium
marko.seslija@gmail.com

Prof. John Shawe-Taylor
Catholic University of Louvain - UCL
5 Chapel Square
GU25 4SZ Virginia Water -
United Kingdom
jst@cs.ucl.ac.uk

Dr. Lei Shi
KU Leuven, ESAT-SCD
Kasteelpark Arenberg 10
3001 Leuven-Heverlee - Belgium
leishi@fudan.edu.cn

Dr. Marco Signoretto
KU Leuven, ESAT-SCD
Kasteelpark Arenberg 10
3001 Leuven-Heverlee - Belgium
marco.signoretto@esat.kuleuven.be

Pascal Spincemaille
Weill Cornell Medical College
416 E55th St
10022 New York - United States
pas2018@med.cornell.edu

Santosh Srivastava
Jaypee University of Engineering
and Technology
Department of Mathematics
Guna, M.P, India
473226 Raghogarh - India
santosh_00147@rediffmail.com

Prof. Johan Suykens
KU Leuven, ESAT-SCD
Kasteelpark Arenberg 10
3001 Leuven-Heverlee - Belgium
johan.suykens@esat.kuleuven.be

Tatiana Tommasi
KU Leuven, ESAT-PSI/VISICS
Kasteelpark Arenberg 10
3001 Leuven-Heverlee - Belgium
tatiana.tommasi@esat.kuleuven.be

Prof. Joel Tropp
Caltech
1200 E. California Blvd. MC 305-16
91125-5000 Pasadena - United States
jtropp@caltech.edu

Dr. Konstantin Usevich
Vrije Universiteit Brussel
Pleinlaan 2, Building K
Department ELEC
1050 Brussels - Belgium
konstantin.usevich@vub.ac.be

Dr. Vanya Van Belle
KU Leuven, ESAT-SCD
Kasteelpark Arenberg 10
3001 Leuven-Heverlee - Belgium
vanya.vanbelle@esat.kuleuven.be

Dr. Laurens van der Maaten
Delft University of Technology
Mekelweg 4
2628 CD Delft - The Netherlands
lvdmaaten@gmail.com

Dr. Steven Van Vaerenbergh
University of Cantabria
Departamento de Ingenieria de
Comunicaciones
Edificio Ingenieria de Telecomunicacion
Plaza de la Ciencia s/n
39005 Santander - Spain
steven@gtas.dicom.unican.es

Prof. Joos Vandewalle
KU Leuven, ESAT-SCD
Kasteelpark Arenberg 10
3001 Leuven-Heverlee - Belgium
joos.vandewalle@esat.kuleuven.be

Carolina Varon
KU Leuven, ESAT-SCD
Kasteelpark Arenberg 10
3001 Leuven-Heverlee - Belgium
Carolina.Varon@esat.kuleuven.be

Yuyi Wang
KU Leuven, Dep. Computer Science
Celestijnenlaan 200A
3001 Leuven-Heverlee - Belgium
Yuyi.Wang@cs.kuleuven.be

Zi Wang
Imperial College London
2 Gainsborough Road
London E15 3AF - United Kingdom
zi.wang11@imperial.ac.uk

Dr. Marco Wiering
University of Groningen
Department of Artificial Intelligence
Nijenborgh 9
9700AK Groningen - The Netherlands
m.a.wiering@rug.nl

Dr. Gunnar Wilken
Okinawa Institute of Science
and Technology
1919-1 Tancha, Lab 1, SCB Unit
904-0495 Onna-son - Japan
wilken@OIST.JP

Ines Wilms
KU Leuven
Onderzoeksgroep Operations Research
and Business Statistics
Naamsestraat 69
3000 Leuven - Belgium
ines.wilms@kuleuven.be

Bahman Yari Saeed Khanloo
CWI
123 Science Park
1098XG Amsterdam - The Netherlands
khanloo@cwi.nl

Pooya Zakeri
KU Leuven, ESAT-SCD
Kasteelpark Arenberg 10
3001 Leuven-Heverlee - Belgium
Pooya.Zakeri@esat.kuleuven.be

Roman Zakharov
Universite catholique de Louvain - UCL
2, Place Sainte-Barbe
A158
1348 Louvain-la-Neuve - Belgium
roman.zakharov@uclouvain.be

Dr. Rafal Zdunek
Wroclaw University of Technology
Wybrzeze Wyspianskiego 27
50-370 Wroclaw - Poland
rafal.zdunek@pwr.wroc.pl

Prof. Ding-Xuan Zhou
City University of Hong Kong
Dep. of Mathematics
83 Tat Chee Avenue, Kowloon
Hong Kong - China
mazhou@cityu.edu.hk