SPAMS (SPArse Modeling Software)

Optimization toolbox for sparse estimation problem resolution

Ghislain DURIF

January 22nd, 2021

IMAG – CNRS

Introduction

Credits

- **Main development**: Julien Mairal (Inria)
- Contributions: Francis Bach (Inria), Jean Ponce (Ecole Normale Supérieure), Guillermo Sapiro (University of Minnesota), Guillaume Obozinski (Inria) and Rodolphe Jenatton (Inria), Yuansi Chen (Inria), Zaid Harchaoui (Inria)
- R and Python interfaces: Jean-Paul Chieze (Inria)
- Development and maintenance of **version >=2.6** (especially porting to R-3.x and Python-3.x): Ghislain Durif

What is SPAMS

SPAMS = SPArse Modeling Software

- An open-source **optimization** toolbox for **sparse estimation**
- Implementation of algorithms to solve machine learning and signal processing problems involving **sparse regularizations**

`http://spams-devel.gforge.inria.fr/`

Status

- Not in active development anymore
- Only **light maintenance**
- Widely used (**10000+ downloads** per year since 2012), even today
- Modern replacement : see Cyanure[1] (Python only for the moment)

[1] `http://thoth.inrialpes.fr/people/mairal/cyanure/welcome.html`

Technical specs

- Implemented in a C++ library (usable in C++ projects)
- Compatible with Linux, Mac (no Windows support in the latest version)
- Interfaced with Matlab[2], R and Python (both automatically generated from the Matlab interface with SWIG[3])
- Python interface is now “autonomous” with a dedicated repository[4]
- Based on efficient linear algebra libray and multi-threading through OpenMP.

[2] historical interface

[3] `http://swig.org/`

[4] `https://gitlab.inria.fr/thoth/python-spams`

Development

- Main project at `https://gitlab.inria.fr/thoth/spams-devel` (version>=2.6)
- Python interface at `https://gitlab.inria.fr/thoth/python-spams` (version>=2.6.1)
- **Legacy project** at `https://gforge.inria.fr/projects/spams-devel` (2.2<=version<=2.5)

Requirements

- An implementation of BLAS and LAPACK to perform linear algebra operations.

Possible to use

- BLAS and LAPACK library shipped with R
- external libraries such as atlas, OpenBlas, or the Intel Math Kernel Library (MKL[5])

Note: the MKL is recommended to get the best performance and is available (even for R[6]) with the Anaconda Python distribution[7]

[5] https://en.wikipedia.org/wiki/Math_Kernel_Library
[6] https://docs.anaconda.com/anaconda/user-guide/tasks/using-r-language/
[7] https://docs.anaconda.com/anaconda/install/

Content

Various toolboxes

- Dictionary learning and matrix factorization
- Sparse decomposition
- Proximal methods

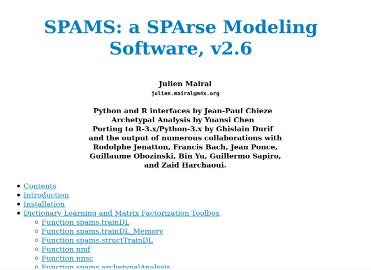
+ **various functions** to perform linear algebra operations such as a conjugate gradient algorithm, manipulating sparse matrices and graphs

Documentation

Available for each interface (R[8], Matlab and Python) at

`http://spams-devel.gforge.inria.fr/documentation.html`

old school but complete



SPAMS: a SPArse Modeling Software, v2.6

Julien Mairal

julien.mairal@m4x.org

Python and R interfaces by Jean-Paul Chieze
Archetypal Analysis by Yuansi Chen
Porting to R-3.x/Python-3.x by Ghislain Durif
and the output of numerous collaborations with
Rodolphe Jenatton, Francis Bach, Jean Ponce,
Guillaume Obozinski, Bin Yu, Guillermo Sapiro,
and Zaid Harchaoui.

[8] R documentation at `http://spams-devel.gforge.inria.fr/doc-R/html/index.html`

Availability and installation

- SPAMS is not available on CRAN.
- The latest version of SPAMS is currently not available for Windows.
- Should be **installed from source**, R package available at the official website[9]

[9] `http://spams-devel.gforge.inria.fr/downloads.html`

Short introduction about optimization

Optimization problem (intuitively)

- Given a function $F : \mathbb{R}^p \rightarrow \mathbb{R}$ called the **objective**
- Aim at finding the point $\mathbf{x}^* \in \mathbb{R}^p$ such that $F(\mathbf{x}^*)$ is **optimal**
- Generally **maximizing** or **minimizing** the objective F
- **Complexity** of the optimization depends on the **characteristics** and **regularity** of F
- **Different algorithms** depending on the **complexity** of the problem

Optimum and optimal argument (intuitively)

- $\mathbf{x}^* \in \mathbb{R}^p$ is an **argmin** and $F(\mathbf{x}^*) \in \mathbb{R}$ a **min** if $F(\mathbf{x}^*) \leq F(\mathbf{x})$ for any $\mathbf{x} \in \mathbb{R}^p$

$$\mathbf{x}^* = \underset{\mathbf{x} \in \mathbb{R}^p}{\operatorname{argmin}} F(\mathbf{x})$$

- $\mathbf{x}^* \in \mathbb{R}^p$ is an **argmax** and $F(\mathbf{x}^*) \in \mathbb{R}$ a **max** if $F(\mathbf{x}^*) \geq F(\mathbf{x})$ for any $\mathbf{x} \in \mathbb{R}^p$

$$\mathbf{x}^* = \underset{\mathbf{x} \in \mathbb{R}^p}{\operatorname{argmax}} F(\mathbf{x})$$

- **local optimum** if optimum on a **bounded region** of $\mathbb{R}^p$

Minimization and maximization

$$\operatorname*{argmax}_{\mathbf{x} \in \mathbb{R}^p} F(\mathbf{x}) = \operatorname*{argmin}_{\mathbf{x} \in \mathbb{R}^p} \; - F(\mathbf{x})$$

Illustration: max and argmax

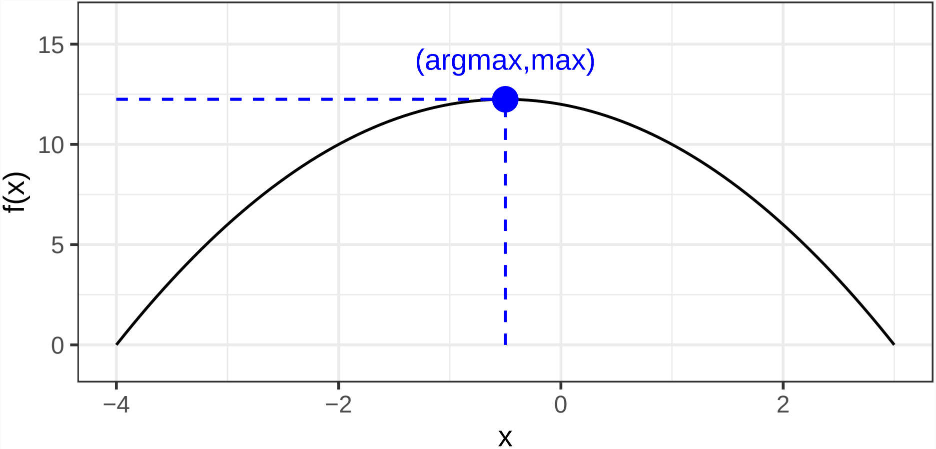
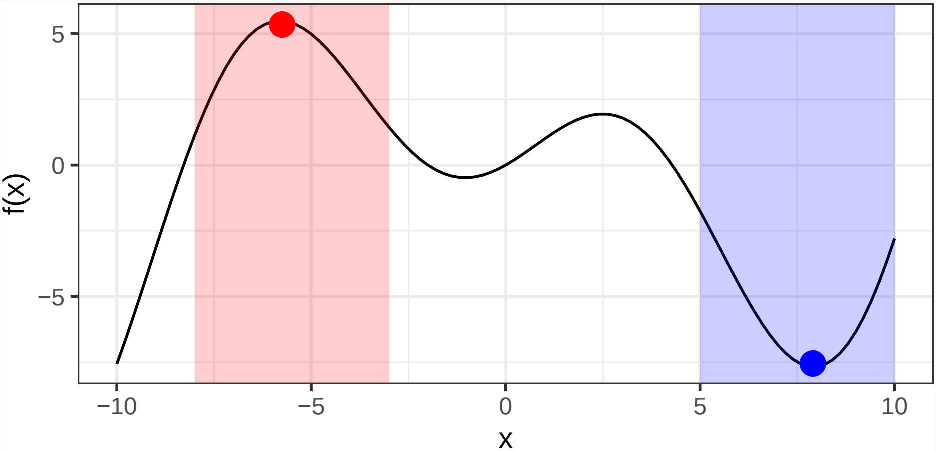


Illustration: local optimum



Optimization problem in statistics and machine learning

Many statistical or learning problems are defined thanks to an optimization problem

- Regression
- Classification
- Dimension reduction (latent space projection, variable selection)
- and more...

Data = collection (sample) of n observations $(\mathbf{x}_i)_{i=1,\ldots,n}$ with $\mathbf{x}_i \in \mathbb{R}^p$

Optimization problem

$$\underset{\mathbf{w} \in \mathbb{R}^p}{\operatorname{argmin}} \left(\sum_{i=1}^{n} f(\mathbf{x}_i, \mathbf{w}) \right) + \lambda\, P(\mathbf{w})$$

- $f : \mathbb{R}^p \times \mathbb{R}^p \longrightarrow \mathbb{R}$ is the **loss** function
- $P : \mathbb{R}^p \longrightarrow \mathbb{R}$ is the **penalty** function
- $\lambda \in \mathbb{R}$ is the penalization **parameter**

Supervised problem

Data = collection (sample) of n observations $\left(\mathbf{x}_i, y_i\right)_{i=1,\dots,n}$ with $\mathbf{x}_i \in \mathbb{R}^p$ and $y_i \in \mathbb{R}$

Optimization problem

$$\underset{\mathbf{w}\in\mathbb{R}^p}{\operatorname{argmin}} \sum_{i=1}^{n} f(\mathbf{x}_i, y_i, \mathbf{w}) + \lambda\, P(\mathbf{w})$$

- $f : \mathbb{R}^p \times \mathbb{R} \times \mathbb{R}^p \longrightarrow \mathbb{R}$ is the **loss** function
- $P : \mathbb{R}^p \longrightarrow \mathbb{R}$ is the **penalty** function
- $\lambda \in \mathbb{R}$ is the penalization **parameter**

Example of linear regression

Model

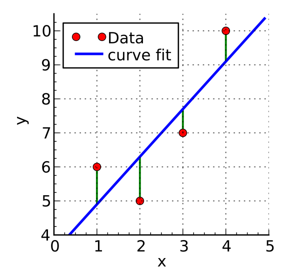
$$Y_i = \beta_0 + \left(\sum_{j=1}^{p} x_{ij}\, \beta_j \right) + \varepsilon_i$$

Optimization problem

$$(\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_p) = \underset{(\beta_0, \beta_1, \ldots, \beta_p) \in \mathbb{R}^{p+1}}{\operatorname{argmin}} \sum_{i=1}^{n} \left(y_i - \left(\beta_0 + \sum_{j=1}^{p} x_{ij}\, \beta_j \right) \right)^2$$

or with matrix notation

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta} \in \mathbb{R}^{p+1}}{\operatorname{argmin}} \left\| \mathbf{y} - \mathbf{X}\, \boldsymbol{\beta} \right\|_2^2$$



Source: `wikimedia.org`

Ridge regularization (ℓ_2 penalty)

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta} \in \mathbb{R}^{p+1}}{\operatorname{argmin}} \left\| \mathbf{y} - \mathbf{X}\,\boldsymbol{\beta} \right\|_2^2 + \lambda \left\| \boldsymbol{\beta} \right\|_2^2$$

Lasso (sparsity-inducing ℓ_1 penalty)

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta} \in \mathbb{R}^{p+1}}{\operatorname{argmin}} \left\| \mathbf{y} - \mathbf{X}\,\boldsymbol{\beta} \right\|_2^2 + \lambda \left\| \boldsymbol{\beta} \right\|_1$$

Many **different** optimization algorithms

- **exact resolution** based on objective properties (very rare, e.g. least squares regression in some cases)
- **iterative approximate resolution** (e.g. coordinate descent, gradient descent, etc.)

The properties (regularity, smoothness, differentiability, etc.) of the objective condition the **difficulty** of the resolution and the **possible** optimization algorithms that can be used.

SPAMS toolboxes

Official documentation

`http://spams-devel.gforge.inria.fr/doc-R/html/index.html`

with bibliography and references

Sparse dictionary learning and matrix factorization (I)

- dictionary Learning for sparse coding sparse principal component analysis (seen as a sparse matrix factorization problem)
- non-negative matrix factorization
- non-negative sparse coding
- dictionary learning with structured sparsity
- archetypal analysis

Sparse dictionary learning and matrix factorization (II)

Data = collection (sample) of n observations $(\mathbf{x}_i)_{i=1,\ldots,n}$ with $\mathbf{x}_i \in \mathbb{R}^p$

Optimization problem

$$\underset{\mathbf{D}\in\mathbb{R}^{p\times K}}{\mathrm{argmin}} \; \frac{1}{n}\sum_{i=1}^{n}\left(\underset{\boldsymbol{\alpha}_i\in\mathbb{R}^K}{\mathrm{argmin}} \; \frac{1}{2}\|\mathbf{x}_i - \mathbf{D}\,\boldsymbol{\alpha}_i\|_2^2 + \lambda\, P(\alpha_i)\right)$$

Various **penalty** functions P and **sub-problems** (optimization on $\mathbf{D}$ or $\boldsymbol{\alpha}_i$)

Sparse decomposition

Various problems and algorithms including:

- Orthogonal Matching Pursuit (or Forward Selection)
- LARS, coordinate descent algorithms
- and more

Example of optimization problem:

$$\underset{\boldsymbol{\alpha}\in\mathbb{R}^K}{\operatorname{argmin}} \frac{1}{2}\|\mathsf{x} - \mathsf{D}\,\boldsymbol{\alpha}\|_2^2 + \lambda\, P(\alpha)$$

with various **penalty** functions P

Proximal toolbox

Resolution of the following problem with proximal methods

$$\underset{\mathbf{w} \in \mathbb{R}^p}{\operatorname{argmin}} \, f(\mathbf{w}) + \lambda \, P(\mathbf{w})$$

- various **loss** functions f including the square loss, logistic loss, multi-class logistic loss
- various **penalty** functions P including ℓ_q norms, Elastic net, Fused Lasso, group Lasso tree structured norms, trace norm, etc.

Implementing **ISTA** and **FISTA** algorithms

And more

- linear algebra operations such as a conjugate gradient algorithm
- manipulating sparse matrices and graphs