SPAMS (SPArse Modeling Software)

Optimization toolbox for sparse estimation problem resolution

Ghislain DURIF

January 22nd, 2021

IMAG – CNRS

Introduction

- **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

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/
```

- **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

- 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 library and multi-threading through OpenMP.

---

[2]historical interface
[3]http://swig.org/
[4]https://gitlab.inria.fr/thoth/python-spams

- 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)

- 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/

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

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

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

old school but **complete**

---

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

- 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

- Given a function $F : \mathbb{R}^p \rightarrow \mathbb{R}$ called the **objective**

- Aim at finding the point $x^* \in \mathbb{R}^p$ such that $F(x^*)$ 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

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

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

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

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

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

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

Illustration: max and argmax
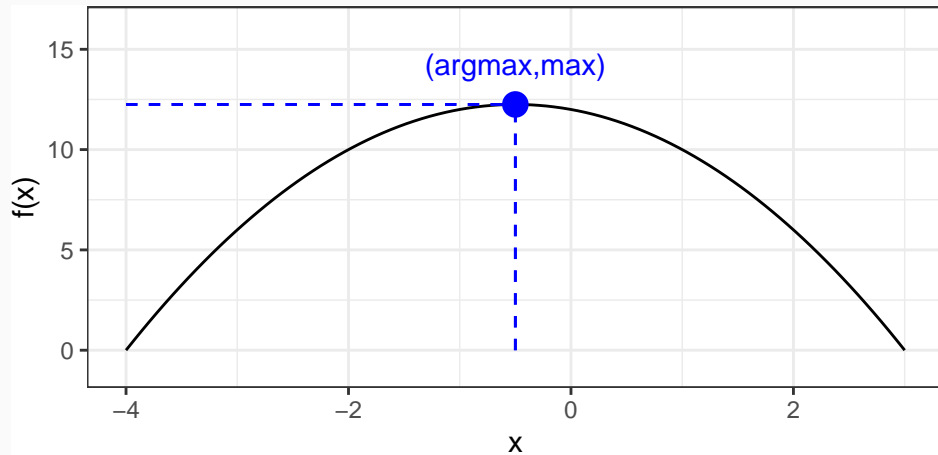
Illustration: local optimum

Optimization problem in statistics and machine learning
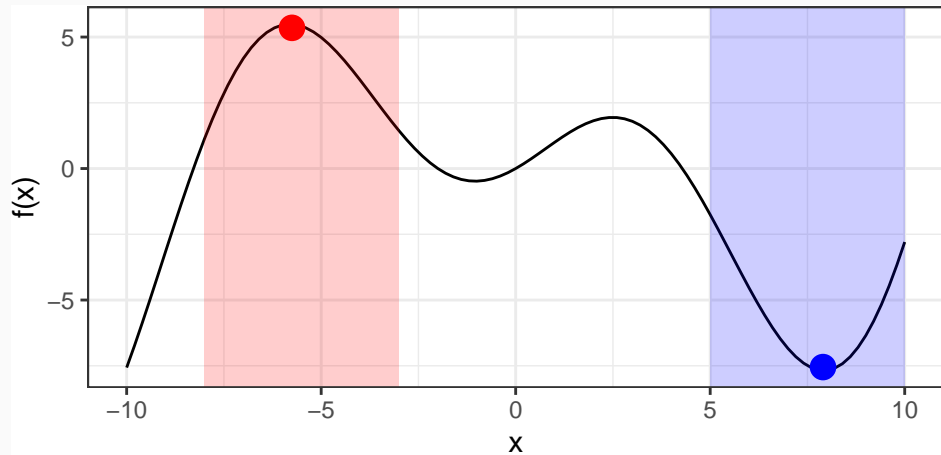
Many statistical or learning problem are defined thanks to an optimization problem

- Regression

- Classification

- Dimension reduction (latent space projection, variable selection)

- and more...

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

Optimization problem

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

- $f : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$ is the **loss** function

- $P : \mathbb{R}^p \to \mathbb{R}$ is the **penalty** function

- $\lambda \in \mathbb{R}$ is the penalization **parameter**

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

Optimization problem

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

- $f : \mathbb{R}^p \times \mathbb{R} \times \mathbb{R}^p \to \mathbb{R}$ is the **loss** function

- $P : \mathbb{R}^p \to \mathbb{R}$ is the **penalty** function

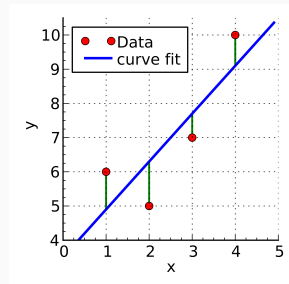- $\lambda \in \mathbb{R}$ is the penalization **parameter**

Model

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

Optimization problem

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

or with matrix notation

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

Source: `wikimedia.org`

Ridge regularization (ℓ_2 penalty)

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

Lasso (sparsity-inducing ℓ_1 penalty)

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

Optimization algorithm in general

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 $(x_i)_{i=1,\dots,n}$ with $x_i \in \mathbb{R}^p$

Optimization problem

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

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

Sparse decomposition

Various problems and algorithms including:

- Orthogonal Matching Pursuit (or Forward Selection)

- LARS, OMP, coordinate descent algorithms

- and more

Example of optimization problem:

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

with various **penalty** functions P

Proximal toolbox

Resolution of the following problem with **proximal methods**

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

- 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

- linear algebra operations such as a conjugate gradient algorithm

- manipulating sparse matrices and graphs