Internship proposal: "Hyperparameter tuning: calibrating MCP and other sparse regression formulations"

Chair of research and teaching in artificial intelligence

Keywords: Optimization, calibration, cross-validation, sparse regression, MCP

Hyperparameter calibration for variable selection

Learning methods in a context where the number of features can be greater than the number of observations have been popularized by genomics applications. They are common for bio-statistics inference, where n the number of patients is limited, while p the number of features available are numerous (*e.g.,* clinical or genetic features). This field has fostered the spread of regularized least-squares with sparsity inducing penalties.[1] More recently, dictionary learning [8] or statistical inference (*e.g.,* to provide faithful confidence intervals [13]) have relied on multiple regressors whose tuning time has become prohibitive. Yet, such regularized methods are sensitive to such tuning parameters, trading-off data fitting and sparsity. Improving tuning procedures is a major concern shared by most practitioners, might it be neuro-imaging or bio-statistics. For practitioners, tuning is a hurdle, and they usually resort to default settings from packaged methods: time constraints discourage them from investigating further settings. Hence, it is crucial to provide high dimensional regression methods that have automatic tuning properties.

The standard tuning procedure in ML is cross-validation: 1) a grid of tuning parameters is created; 2) the dataset is divided into equal sized subsamples (folds), and alternatively each fold is left aside as a validation set, while training (for each parameter) is performed on the other folds. 3) A score aggregates the validation feedback and the parameter achieving the best one is selected. Though popular in practice, cross-validation has important drawbacks: **1) theoretical** efficiency analysis in high dimension is scarce [1]; **2) computational** inefficiency: for 100 values of a 1D parameter, 10-fold cross-validation requires evaluating 1000 estimators, and is unpractical for more than 3 parameters.

Tuning procedures satisfying jointly statistical and computational efficiency are so far still missing, despite recent advances [5, 11]. In a preliminary work with A. Gramfort and colleagues, we have shown that **bi-level** optimization techniques could help to handle many hyper-parameters to be tuned for Lasso-like methods [3]. Leveraging **automatic differentiation** and the underlying convex structure of the regressors, we have managed to efficiently tune p parameters, *i.e.,* one per feature. Early results (see `sparse-ho`) have been obtained for standard left-out strategies, but vanilla cross-validation is not yet handled. A deeper investigation is required to generalize our strategy to more diverse learning frameworks.

Internship description

Let us remind the definition of the Minimax Concave Penalty (MCP) estimator. First, for some parameters $\gamma > 1$ and $\lambda \geq 0$, we define the 1D penalty for any $t \in \mathbb{R}$ as follows:

[1] Sparsity is key as it allows practitioners to interpret the influence of each feature on their models.

$$p^{\text{MCP}}_{\lambda,\gamma}(t) = \begin{cases} \lambda|t| - \frac{t^2}{2\gamma}, & \text{if } |t| \leq \gamma\lambda, \\ \frac{1}{2}\gamma\lambda^2, & \text{if } |t| > \gamma\lambda \ . \end{cases} \tag{1}$$

The proximity operator[2] of $p_{\lambda,\gamma}$ for parameters $\lambda > 0$ and $\gamma > 1$ is defined as follows (see [4, Sec. 2.1]):

$$\text{prox}^{\text{MCP}}_{\lambda,\gamma}(t) = \begin{cases} \frac{\text{ST}(t,\lambda)}{1-\frac{1}{\gamma}} & \text{if } |t| \leq \gamma\lambda \\ t & \text{if } |t| > \gamma\lambda \ , \end{cases} \tag{2}$$

where $\text{ST}(t,\lambda) = \text{sign}(t) \cdot (|t| - \lambda)_+$ for any $t \in \mathbb{R}$ and $\lambda \geq 0$, and similarly $\text{HT}(t,\lambda) = t \cdot \mathbb{1}_{\{|t|>\lambda\}}$. The proximal operator defined in Eq.(2) is also known as the Firm Shrinkage [6]. The MCP estimator is then given by:

$$\hat{\beta}^{(\lambda,\gamma)}(y) \triangleq \underset{\beta\in\mathbb{R}^p}{\arg\min} \frac{1}{2n}\|y - X\beta\|_2^2 + \sum_{j=1}^{p} p^{\text{MCP}}_{\lambda,\gamma}(\beta_j) \ , \tag{3}$$

where $X \in \mathbb{R}^{n\times p}$ is the design matrix and $y \in \mathbb{R}^n$ the observed signal.

Illustration of the penalty and proximal operator are provided in Fig.1 for some parameters, including limiting behavior.

The optimization solvers of interest that will be investigated are the following (the first one could be enough for a first step):

- Coordinate descent [12, 4]
- Proximal gradient descent [2]
- Difference of convex (DC) programming [7]

An efficient implementation allowing to select the parameter efficiently (say using cross-validation) will be of high interest. Early attempts using clever grid search [9] or optimization [3] will be evaluated in terms of time and statistical performance.

Alternative measures of performance for the tuning step might also be investigated, including randomized procedures. Of interest could be variants where the hyperparameter is optimized using a random train/test split (tuning with the test part) for different replicas. Then, a stability selection [10] procedure could help performing variable selection.

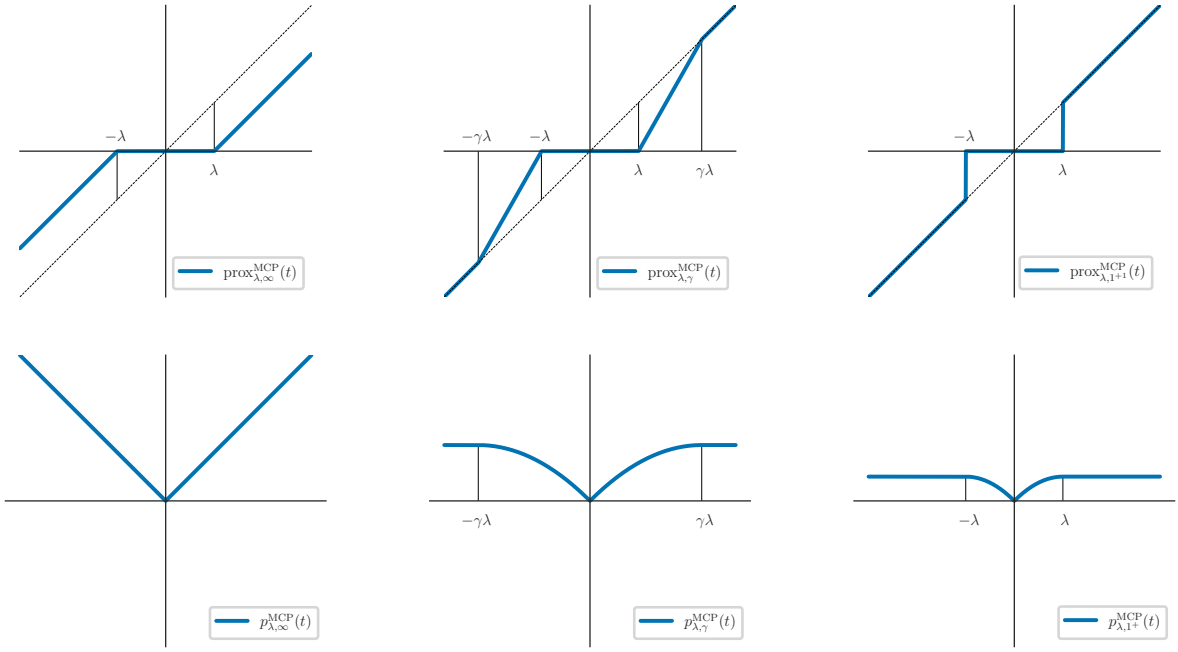
Skills required

- `Python`
- `Git`
- `R` (not mandatory, but could come handy)

Supervision Team

- Joseph Salmon: joseph.salmon@umontpellier.fr
- Cássio Fraga Dantas.: cassiofragadantas@gmail.com
- Emmanuel Soubies: Emmanuel.Soubies@irit.fr

[2]For a function p this operator is defined by $\text{prox}(x) = \arg\min_{x'} p(x') + \|x - x'\|^2/2$.



(a) $\gamma \to +\infty$: recovers $\mathrm{ST}(\cdot, \lambda)$ (b) $\gamma = 2.3$ (c) $\gamma \to 1^+$: recovers $\mathrm{HT}(\cdot, \lambda)$

Figure 1: Penalty (bottom) and associated proximal operator (top) for a fixed λ, with $\gamma \to +\infty$ (a), for $\gamma = 2.3$ (b) and $\gamma \to 1+$ (c).

Salary

Gross monthly salary: approx 550 Euros.
This work will be funded by the ANR CaMeLOt ANR-20-CHIA-0001-01.

Duration

The internship could last from 4 to 6 months.

Location

The internship will be located in Montpellier (Univ. Montpellier), within the mathematics department (IMAG).

References

[1] S. Arlot and A. Celisse. “A survey of cross-validation procedures for model selection”. *Statistics surveys* 4 (2010), pp. 40–79.

[2] A. Beck and M. Teboulle. “A fast iterative shrinkage-thresholding algorithm for linear inverse problems”. *SIAM J. Imaging Sci.* 2.1 (2009), pp. 183–202.

[3] Q. Bertrand et al. “Implicit differentiation of Lasso-type models for hyperparameter optimization”. *ICML*. 2020.

[4] P. Breheny and J. Huang. “Coordinate descent algorithms for nonconvex penalized regression, with applications to biological feature selection”. *Ann. Appl. Stat.* 5.1 (2011), p. 232.

[5] D. Chételat, J. Lederer, and J. Salmon. “Optimal two-step prediction in regression”. *Electron. J. Stat.* 11.1 (2017), pp. 2519–2546.

[6] H.-Y. Gao and A. G. Bruce. “WaveShrink with firm shrinkage”. *Statist. Sinica* (1997), pp. 855–874.

[7] G. Gasso, A. Rakotomamonjy, and S. Canu. “Recovering sparse signals with non-convex

penalties and DC programming". *IEEE Trans. Signal Process.* 57.12 (2009), pp. 4686–4698.

[8] J. Mairal, F. Bach, and J. Ponce. "Task-driven dictionary learning". *IEEE Trans. Pattern Anal. Mach. Intell.* 34.4 (2012), pp. 791–804.

[9] R. Mazumder, J. H. Friedman, and T. Hastie. "Sparsenet: Coordinate descent with nonconvex penalties". *J. Amer. Statist. Assoc.* 106.495 (2011), pp. 1125–1138.

[10] N. Meinshausen and P. Bühlmann. "Stability selection". *J. R. Stat. Soc. Ser. B Stat. Methodol.* 72.4 (2010), pp. 417–473.

[11] E. Ndiaye et al. "Safe Grid Search with Optimal Complexity". *ICML*. Vol. 97. 2019, pp. 4771–4780.

[12] T. T. Wu and K. Lange. "Coordinate descent algorithms for lasso penalized regression". *Ann. Appl. Stat.* (2008), pp. 224–244.

[13] C.-H. Zhang and S. S. Zhang. "Confidence intervals for low dimensional parameters in high dimensional linear models". *J. R. Stat. Soc. Ser. B Stat. Methodol.* 76.1 (2014), pp. 217–242.