AMP for SLOPE

Bartłomiej Polaczyk

19 July 2020

Abstract

The aim of this project is to investigate the the approximate message passing algorithm for SLOPE regularization problem based on (Bu et al. 2019) and compare it with classical convex optimization methods. Some numerical experiments regarding the cases that do not fit into the theoretical framework of (Bu et al. 2019) are also performed and analyzed.

Theoretical bacground

Introduction

We are interested in solving the standard linear inverse problem

$$y = Ax + w, \tag{1}$$

where $y \in \mathbb{R}^n$ and $A \in \mathbb{R}^{n \times p}$ are known parameters of the model, $w \in \mathbb{R}^n$ is a random noise vector and $x \in \mathbb{R}^p$ is an unknown vector of paramteres we wish to estimate. We assume $p \gg n$, i.e. the number of features is much greater than the size of the sample data and whence there might be many potential solutions to the problem (1).

To resolve this issue and prevent overfitting, we introduce a penalty function ϕ which faforizes sparse solutions of (1), i.e. now we are looking among the minimizers of the following form

$$\widehat{x} = \underset{x}{\operatorname{argmin}}\{\, \frac{1}{2}\|Ax - y\|_2^2 + \phi(x)\,\}. \tag{2}$$

The usual choices of ϕ are scaled l^2 penalty (Tikhonov regularization) and l^1 penalty (LASSO). This note concerns sorted l^1 penalized estimation (abbrev. SLOPE), introduced for the first time in (Bogdan et al. 2015), which assumes ϕ to be the sorted l^1 penalty, i.e.

$$\phi(x) = \phi_\lambda(x) = \sum_{i=1}^{n} \lambda_i x_i^\downarrow,$$

where $x_1^\downarrow \geq x_2^\downarrow \geq \ldots \geq x_n^\downarrow$ is the ordered permutation of the vector $|x| = (|x_1|, |x_2|, \ldots, |x_n|)$ and $\lambda_1 \geq \lambda_2 \geq \ldots \lambda_p \geq 0$ are hyperparameters of the model. To lighten notation, we denote $\Delta_p = \{\, x \in \mathbb{R}^p \colon x_1 \geq x_2 \geq \ldots \geq x_p \geq 0\,\}$, so that the above requirements read: $x^\downarrow, \lambda \in \Delta_p$, where $x^\downarrow = P\,|x|$ for some permutation P of the set $\{1, 2, \ldots, p\}$. Such a choice of regulizer is a generalization of the l^1 regularization, as can be seen by taking $\lambda_i \equiv \text{const}$.

The fact that ϕ_λ is non-separable makes the analysis of its teoretical properties much more onerous than in case of classical (separable) models, cf. (Bogdan et al. 2015, Bu et al. (2019)). Nonetheless, it turns out that SLOPE has two advantages over other regularization methods such as LASSO and knocoffs, namely:

1. it achieves certain minimax estimation properties under particular random designs *without* requiring any knowledge of the sparsity degree of $\widehat{x}$, cf. (Bellec, Lecué, and Tsybakov 2018);
2. it controls the false discovery rate in the case of independent predictors, cf. (Bogdan et al. 2015).

We are interested in the algorithmic solutions to the problem (2). Since the objective function in (2) is convex but not smooth, one can not apply directly the classical gradient descent and has to turn to other methods.

A natural alternative solution is the plethora of proximal algorithms, e.g. ISTA (and its improvement – FISTA, cf. (A. Beck and Teboulle 2009)) or more classical alternating direction methods of multipliers (ADMM). The methods have been throughly studied in the literature, cf. (Beck A 2017) for a detailed treatment of the subject.

In this note we will focus on another approach, via the approximate message passing, considered for the first time in context of the LASSO problem in (Donoho, Maleki, and Montanari 2009) and subsequentially developed in e.g. (Bayati and Montanari 2011), and for the SLOPE regularization in (Bu et al. 2019) – see e.g. (Zdeborová and Krzakala 2016) for an accessible derivation of the method.

In the subsequent sections we will describre briefly some of these approaches.

Proximal methods

Denoting $g(x) = \frac{1}{2}\|Ax-y\|_2^2$, the iterative shrinkage thresholding algorithm (ISTA) iteration for SLOPE with given λ can be written as:

ISTA-SLOPE: Input: $y \in \mathbb{R}^p$, $\lambda \in \Delta_p$, $A \in \mathbb{R}^{n\times p}$ Initialize $g(x) = \frac{1}{2}\left\|Ax-y\right\|^2$, $x \in \mathbb{R}^p$, $t > 0$ **while** (*stopping condition*) **{** $x \leftarrow \operatorname{prox}_{t\phi_\lambda}\big(x - t\nabla g(x)\big)$; **} return** x

where t can be thought of as the learning rate and prox denotes the proximal operator given by

$$\operatorname{prox}_h(y) := \underset{x}{\operatorname{argmin}}\{\, h(x) + \frac{1}{2}\|x-y\|_2^2\,\}.$$

A. Beck and Teboulle (2009) have introduced a faster version of ISTA, a.k.a. FISTA, which is based on the idea of Nesterov momentum. The general form of the algorithm is the following:

FISTA-SLOPE: Input: $y \in \mathbb{R}^p$, $\lambda \in \Delta_p$, $A \in \mathbb{R}^{n\times p}$ Initialize $g(x) = \frac{1}{2}\left\|Ax-y\right\|^2$, $x = x_{old} \in \mathbb{R}^p$, $r = 1$, $t > 0$ **while** *(stopping condition)* **{** $u \leftarrow x_{old} + r(x - x_{old})$; $x_{old} \leftarrow x$; $x \leftarrow \operatorname{prox}_{t\phi_\lambda}\big(u - t\nabla g(u)\big)$; *update*$(r)$; **} return** x

Here t can be thought of as a acceleration term, which (if updated correctly through the update rule) can increase substantialy the speed of convergence of the algorithm. Note that keeping $r \equiv 1$ restores ISTA.

One of the difficulties in dealing with SLOPE is that the regulizer ϕ_λ is non-separable and thus its proximal operator cannot be applied element-wise. (Bogdan et al. 2015) have proposed an efficient algorithm that for any $u \in \mathbb{R}^p$ computes

$$\widehat{x} = \operatorname{prox}_{\phi_\lambda}(u) = \underset{x}{\operatorname{argmin}}\left\{\frac{1}{2}\left\|u-x\right\|^2 + \sum_i \lambda_i x_i^{\downarrow}\right\} = \underset{x}{\operatorname{argmin}}\left\{\sum_i \left[\frac{1}{2}(x_i^{\downarrow})^2 + x_i^{\downarrow}\lambda_i - x_i u_i\right]\right\}$$

It is based on the following simple observations that follow immediatly from the above formulation:

1. $\operatorname{sign}(\widehat{x}_i) = \operatorname{sign}(u_i)$ for each i such that $\widehat{x}_i \neq 0$;
2. $P\widehat{x} = \operatorname{prox}_{\phi_\lambda}(Pu)$ for any permutation P;
3. If $u \in \Delta_p$, then $\widehat{x} \in \Delta_p$ (i.e., $u = u^{\downarrow} \Rightarrow \widehat{x} = \widehat{x}^{\downarrow}$);

Therefore we can and do assume in the analysis below that $u \in \Delta_p$. The optimization procedure now reads:

$$\widehat{x} = \underset{x^\downarrow \in \Delta_p}{\operatorname{argmin}} \big\{ \sum_i \big[\frac{1}{2}(x_i^\downarrow)^2 - x_i^\downarrow (u-\lambda)_i \big] \big\},$$

whence

4. $\widehat{x}$ depends only on the vector $(\lambda - u)$;
5. If $(u-\lambda)_+ \in \Delta_p$ then $\widehat{x} = (u-\lambda)_+$, where $v_+ = (\max(v_i, 0))_i$;
6. If $(u-\lambda)_i \leq (u-\lambda)_{i+1}$, then $\widehat{x}_i \leq \widehat{x}_{i+1}$, whence $\widehat{x}_i = \widehat{x}_{i+1}$;
7. Consequently, if $(u-\lambda)$ is nondecreasing along the indices $(i, i+1, \ldots, j)$, then $\widehat{x}_i = \widehat{x}_{i+1} = \ldots = \widehat{x}_{i+1}$ and

   $$\sum_{i \leq r \leq j} \widehat{x}_r^\downarrow (u-\lambda)_r = \sum_{i \leq r \leq j} \widehat{x}_r^\downarrow (\bar{u} - \bar{\lambda})$$

   where $\bar{u}, \bar{\lambda}$ are the means of u and λ along the indices $(i, i+1, \ldots, j)$, so replacing u_r, λ_r with $\bar{u}, \bar{\lambda}$ does not change $\widehat{x}$.

The above observations justify the procedure below proposed by (Bogdan et al. 2015).

FastProxSL1: Input: $u \in \mathbb{R}^p$, $\lambda \in \Delta_p$ # Define the operator $H_u(v) = P(\mathrm{sign}(u_i)v_i)_i$ for some permutation P, so that $H_u(u) = u^\downarrow \in \Delta_p$ $u' \leftarrow H_u(u)$; **while** $(u'-\lambda)_+ \notin \Delta_p$ **{** identify nondecreasing and nonconstant segments $i : j$ of $(u'-\lambda)$ replace u'_r, λ_r for $r \in \{i, i+1, \ldots, j\}$ by their averages $\bar{u'}, \bar{\lambda}$ **} return** $H_u^{-1}(u'-\lambda)_+$;

Approximate message passing

The AMP approach is Bayes in nature. Namely, we assume that the true values of x are i.i.d. from some apriori distribution $X = (X_1, \ldots, X_p)$ satisfying some intregrability choditions and that the noise vector w is elementwise i.i.d. with zero mean and variance $\sigma_w^2 < \infty$. The apriori distribution X and the parameter σ_w^2 will play a crucial role in the construction of the algorithm as will be seen in a while. The base of the AMP algorithm for SLOPE is as follows

AMP-SLOPE: Input: $y \in \mathbb{R}^p$, $\lambda \in \Delta_p$, $A \in \mathbb{R}^{n \times p}$ Initialize: $g(x) = \frac{1}{2}\left\|Ax - y\right\|^2$, $x = x_{old} \in \mathbb{R}^p$, $v = \nabla g(0)$, $t = t(X) \in \mathbb{R}_+$ **while** *(stopping condition)* **{** $x_{old} \leftarrow x$; $x \leftarrow \mathrm{prox}_{\phi_{t\lambda}}\big(x_{old} - v\big)$; $v \leftarrow \nabla g(x) - \frac{v}{n}[\nabla \mathrm{prox}_{\phi_{t\lambda}}(x_{old} - v)]$; *update*$(t)$; **} return** x

Based on the observations 1.-7. above, it is easy to verify that for any vector $u \in \mathbb{R}^p$

$$\nabla \mathrm{prox}_{\phi_\lambda}(u) = \|\mathrm{prox}_{\phi_\lambda}(u)\|_0^*, \quad \text{where} \quad \|u\|_0^* := \#\{\text{unique non-zero magintudes in } |u|\}.$$

E.g., $\|(0,3,-3,3,1)\|_0^* = 2$. Therefore, v update in the AMP-SLOPE algorithm can be read as

$$v \leftarrow \nabla g(x) - \frac{v}{n}\left\|x\right\|_0^*.$$

Moreover, the scaling factor t and its update rule in the AMP scheme are dictated by the so-called state evolution equation. Namely, let

$$F(\tau, \alpha) = \sigma_w^2 + \frac{1}{n}\mathbb{E}\left\|\mathrm{prox}_{\phi_{\tau\alpha}}(X + \tau Z) - X\right\|^2, \quad \tau \in \mathbb{R}_+, \, \alpha \in \Delta^p,$$

where Z is the $\mathcal{N}(0, I_p)$ random vector independent on the whole model. Then, setting

$$\tau_0 = \sigma_w^2 + \frac{p}{n}\mathbb{E}X_1^2, \quad \tau_{k+1}^2 = F(\tau_k, \alpha), \quad \tau_* = \lim_k \tau_k \tag{3}$$

we choose $\hat{\alpha}$ to be the solution of

$$\lambda = \alpha\tau_*\big(1 - \frac{1}{n}\mathbb{E}\|\mathrm{prox}_{\phi_{\tau_*\alpha}}(X + \tau_* Z)\|_0^*\big) \tag{4}$$

and finally

$$t = \frac{\tau}{\tau^*\big(1 - \frac{1}{n}\mathbb{E}\|\mathrm{prox}_{\phi_{\tau_*\hat{\alpha}}}(X + \tau^* Z)\|_0^*\big)}.$$

Bu et al. (2019) have shown under some technical assumptions that there is a unique fixed point of the state evolution (3) (i.e., that τ^* is well defined), and that the solution $\hat{\alpha}$ to (4) is unique and well-defined. They have also proposed a numerical scheme for calculating $\hat{\alpha}$ based on the bisection method which we will exploit.

Numerical experiments

References

Bayati, M., and A. Montanari. 2011. "The Dynamics of Message Passing on Dense Graphs, with Applications to Compressed Sensing." *IEEE Trans. Inform. Theory* 57 (2): 764–85. doi:10.1109/TIT.2010.2094817.

Beck, A., and M. Teboulle. 2009. "A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems." *SIAM J. Imaging Sci.* 2 (1): 183–202. doi:10.1137/080716542.

Beck, A. 2017. *First-Order Methods in Optimization*. Vol. 25. MOS-Siam Series on Optimization. Society for Industrial; Applied Mathematics (SIAM), Philadelphia, PA; Mathematical Optimization Society, Philadelphia, PA. doi:10.1137/1.9781611974997.ch1.

Bellec, P. C., G. Lecué, and A. B. Tsybakov. 2018. "Slope Meets Lasso: Improved Oracle Bounds and Optimality." *Ann. Statist.* 46 (6B): 3603–42. doi:10.1214/17-AOS1670.

Bogdan, M., E. van den Berg, C. Sabatti, W. Su, and E. J. Candès. 2015. "SLOPE—adaptive Variable Selection via Convex Optimization." *Ann. Appl. Stat.* 9 (3): 1103–40. doi:10.1214/15-AOAS842.

Bu, Z., J. Klusowski, C. Rush, and W. Su. 2019. "Algorithmic Analysis and Statistical Estimation of Slope via Approximate Message Passing." In *Advances in Neural Information Processing Systems*, 9366–76. http://par.nsf.gov/biblio/10163278.

Donoho, D., A. Maleki, and A. Montanari. 2009. "Message-Passing Algorithms for Compressed Sensing." *Proceedings of the National Academy of Sciences* 106 (45). National Acad Sciences: 18914–9. doi:https://doi.org/10.1073/pnas.0909892106.

Zdeborová, L., and F. Krzakala. 2016. "Statistical Physics of Inference: Thresholds and Algorithms." *Advances in Physics* 65 (5). Taylor & Francis: 453–552. doi:https://doi.org/10.1080/00018732.2016.1211393.