The MCP: a case study in non convex penalized regression

December 16, 2014

1 Introduction

In the context of least square regression, it has been proven that, under certain circumstances, non convex penalties are more effective for feature selection than convex ones. However, the resulting optimization is challenging due to the non-convexity and non-differentiability of the penalty term. In this paper we compare different local optimization techniques to solve this non convex problem and provide recommendations for practical implementation and use. To this end, we chose to work with the non convex minimum concave penalty (MCP) introduced by Zhang [2010]. Empirical evidence reported confirms the importance of initialization in local optimization procedures.

2 The MCP regression

2.1 The MCP cost function

Let X be an $n \times p$ design matrix and $y \in \mathbb{R}^n$ a response vector given by the following linear model with noise $\varepsilon \in \mathbb{R}^n$

$$Y = X\beta + \varepsilon, \qquad \beta \in \mathbb{R}^p. \tag{1}$$

The MCP estimator $\hat{\beta}_{MCP}$ is given by the minimum of the following MCP cost function for a given couple $(\lambda \geq 0, \gamma \geq 1)$ of hyper parameters:

$$J_{\text{MCP}}(\beta) \;=\; \tfrac{1}{2}\|y - X\beta\|^2 + \sum_{j=1}^{p} \mathbf{pen}_{\lambda,\gamma}(|\beta_j|)\ ,$$

where $\mathbf{pen}_{\lambda,\gamma}$ is a penalty function defined as

$$\mathbf{pen}_{\lambda,\gamma}(t) = \lambda \int_0^t \left(1 - \frac{x}{\gamma\lambda}\right)_+ dx = \begin{cases} \lambda t - \dfrac{t^2}{2\gamma} & \text{if } t \leq \gamma\lambda \\ \dfrac{\gamma\lambda^2}{2} & \text{else.} \end{cases}$$

Parameter λ controls the tradeoff between the loss function and penalty, while parameter γ controls the shape of the penalty as shown figure 1. Note that the MCP penalty can be decomposed as the difference of two convex functions as follows:

$$J_{\text{MCP}}(\beta) = \tfrac{1}{2}\|y - X\beta\|^2 + \lambda\|\beta\|_1 - \lambda \sum_{j=1}^{p} h_{\lambda,\gamma}(|\beta_j|), \tag{2}$$

with

$$h_{\lambda,\gamma}(t) = \left\{ \frac{t^2}{2\gamma\lambda} \mathbb{1}_{\{t \leq \gamma\lambda\}} + \left(t - \frac{\gamma\lambda}{2}\right) \mathbb{1}_{\{t > \gamma\lambda\}} \right\},$$

the Huber penalty function with parameter $\gamma\lambda$, illustrated figure 1.

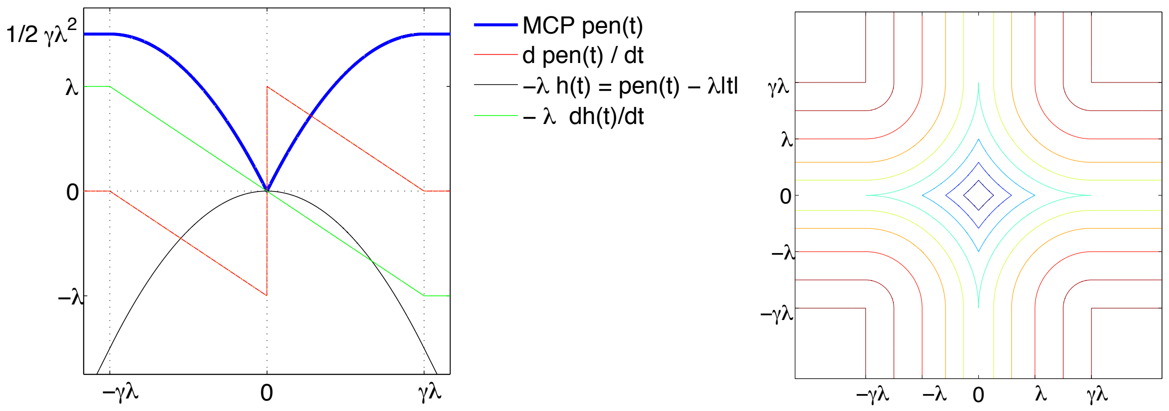

Figure 1: Left: Fonction $\mathbf{pen}_{\lambda,\gamma}(t)$ (blue) for $\lambda = 1$ and $\gamma = 3$, its derivative (red), the associated inverse of the huber loss (black) and its derivative (green). Right: the MCP penalty level set for $\lambda = \gamma = 2$.

2.2 The MCP cost function as a QP

For all λ there exists a positive scalar k such a stationary point of the following quadratic program

$$\begin{cases} \min\limits_{\beta \in \mathbb{R}^p} & \frac{1}{2}\|y - X\beta\|^2 - \lambda \sum_{j=1}^{p} h_{\lambda,\gamma}(|\beta_j|) \\ \text{subject to} & \|\beta\|_1 \leq k \end{cases}$$

is also a stationary point of the MCP cost function. The convexity of this reformulation depends on the positiveness of all the associated Hessian matrices

$$\nabla^2_\beta J_{\text{MCP}}(\beta) \;=\; X^\top X - \frac{1}{\gamma}\,\mathbf{diag}(\mathbf{r}),$$

where $\mathbf{r}$ is a $\mathbb{R}^p$ vector whose components are 1 if $\beta_j \leq \gamma\lambda$ and 0 else, for all β. Zhang [2010] note that the convexity of the square loss overcomes the concavity of the Huber function so that the MCP minimization problem is convex when $\gamma > \frac{1}{c^\star}$, $c^\star$ being the smallest eigen value of matrix $X^\top X$. This is nether the case when p is larger than n. Table 1 illustrates that this is also rarely the case when dealing with real data.

Table 1: Example of γ minimum value guaranteeing the convexity of the MCP on some real data sets from Hastie et al. [2009].

	Housing	Countries	Galaxy	L.A. Ozone	Ozone	Prostate cancer
$\frac{1}{c^\star}$	15.75	1.18e+31	17.14	27.05	3.71	6.41

2.3 The subdiffential of the MCP cost function

Because of the L_1 norm, the loss function $J_{\text{MCP}}(\beta)$ is not differentiable and may be not convex either. Due to the non convexity, the subdiffential defined in the convex case no longer applies. In this case (minimization of a non smooth and non convex functional), if β^* is a local minima of the proper loss function $J_{\text{MCP}}(\beta)$ then it verifies the stationarity condition

$$0 \in \partial_c J_{\text{MCP}}(\beta^*),$$

where $\partial_c J_{\text{MCP}}(\beta^*)$ denotes the Clarke subdifferential of the loss function at point β^* [Clarke, 1990, Bagirov et al., 2014]. The Clake subdifferential is the generalization of the subdifferential to non convex functions. It is defined for locally Lipschitz functions f as the convex hull of some generalized gradient and more precisely

$$\partial_c f(\beta^*) = \left\{ g \in \mathbb{R}^p \;\middle|\; g^\top d \leq D_c f(\beta^*, d) \text{ for all } d \in \mathbb{R}^p \right\}$$

where $D_c f(\beta, d)$ denotes the Clark directional derivative function of f at point β in direction d defined by

$$D_c f(\beta, d) = \limsup_{\epsilon \to 0_+, \delta \to \beta} \frac{f(\delta + \epsilon d) - f(\delta)}{\epsilon}.$$

Note that for J_{MCP} it coincide with the usual directional derivative. For the calculation of $\partial_c J_{\text{MCP}}$ it is worth noticing that equation (2) splits the MCP cost function as the sum of a two terms $J_{\text{MCP}}(\beta) = \phi(\beta) + \varphi(\beta)$, $\phi(\beta) = \frac{1}{2}\|y - X\beta\|^2 - \lambda \sum_{j=1}^{p} h_{\lambda,\gamma}(|\beta_j|)$ being strictly differentiable and $\varphi(\beta) = \lambda \|\beta\|_1$ being convex. In this case [see Clarke, 1990, proposition 2.3.3 corollary 1] the $0 \in \partial_c J_{\text{MCP}}(\beta^*)$ condition comes out as

$$\beta^* \text{ is a local minima} \quad \Rightarrow \quad -\nabla_\beta \phi(\beta^*) \in \partial \varphi(\beta^*) \tag{3}$$

where $\nabla_\beta \phi(\beta^*)$ denotes the gradient of function ϕ at point β^* and $\partial\varphi(\beta^*)$ is the subdifferential of the convex function φ at point β^* (coinciding with its Clarke subdifferential).

2.4 Stationarity conditions for the MCP optimization problem

The Clarke sub differential of the MCP cost function at some β is the vector set of

$$\partial_c J_{\text{MCP}}(\beta) \;=\; X^\top (X\beta - y) + \lambda \begin{cases} \alpha_j & \text{if } \beta_j = 0 \\ \text{sign}(\beta_j) - \dfrac{\beta_j}{\lambda\gamma} & \text{if } |\beta_j| \leq \lambda\gamma \\ 0 & \text{else} \;, \end{cases} \tag{4}$$

for all $\alpha_j \in [-1, 1]$. The details of the component sub gradients are given table 2 So that stationarity conditions are:

$$\begin{cases} |g_j| - \lambda \;\leq\; 0 & \text{if } \beta_j = 0 \\ g_j + \lambda\, \text{sign}(\beta_j) - \dfrac{\beta_j}{\gamma} \;=\; 0 & \text{if } |\beta_j| \leq \lambda\gamma \\ g_j \;=\; 0 & \text{else} \;, \end{cases} \tag{5}$$

with $g = X^\top (X\beta - y)$ the gradient of the square error component of the cost. Note that when the MCP cost is not convex, it may exists various stationary points and local algorithms are only guaranteed to converge towards a local minimum.

3 Algorithms for MCP

This section briefly reviews various local algorithms for solving the MCP minimization problem including the homotopy method (a regularization path algorithm [Zhang, 2010]), subgradient descent, component wise (or shooting, or coordinate descent, or Gauss Seidel [Breheny and Huang, 2011]), multi-stage convex relaxation (or DC [Gasso et al., 2009], or LLA, or CCCP, or MM, or reweighed adaptive lasso [Zhang et al., 2012]) and proximal algorithms with MCP proximal and L_1 proximal [Gong et al., 2013].

Table 2: The components of the MCP cost function and their sub gradients for $\alpha_j \in [-1, 1]$.

	least square error	L_1 penalty	Huber function h
$f(\beta)$	$\frac{1}{2}\|y - X\beta\|^2$	$\lambda\|\beta\|_1$	$\lambda \sum_{j=1}^{p} \begin{cases} \dfrac{\beta_j^2}{2\gamma\lambda} & \text{if } \|\beta_j\| \leq \gamma\lambda \\ \|\beta_j\| - \dfrac{\gamma\lambda}{2} & \text{else} \end{cases}$
$\partial_\beta f(\beta)$	$X^\top(X\beta - y)$	$\lambda \begin{cases} \alpha_j & \text{if } \beta_j = 0 \\ \text{sign}(\beta_j) & \text{else} \end{cases}$	$\lambda \begin{cases} \dfrac{\beta_j}{\gamma\lambda} & \text{if } \|\beta_j\| \leq \gamma\lambda \\ \text{sign}(\beta_j) & \text{else} \end{cases}$

3.1 The homotopy method for MCP

Assume that, For a given λ, the three sets I_0, I_P, I_N where respectively $\beta_j = 0, |\beta_j| \leq \gamma\lambda$ and $|\beta_j| > \gamma\lambda$ are given. Under these hypothesis, the non zero components $\beta_I = (\beta_P^\top, \beta_N^\top)^\top$ of vector β, are given by solving the following linear system (after some reordering)

$$\left(X_I^\top X_I - \frac{1}{\gamma}\,\mathbf{diag}(\mathbf{r}_I) \right) \beta_I(\lambda) = X_I^\top y - \lambda \begin{pmatrix} sign(\beta_P) \\ 0_N \end{pmatrix}, \qquad (6)$$

with $\mathbf{r}_I = \begin{pmatrix} 1\!\!1_P \\ 0_N \end{pmatrix}$, $1\!\!1_P$ being a vector of ones of size $\#I_P$, $I = I_P \cup I_N$ and 0_N being a vector of zeros of size $\#I_N$.

Considering λ' in a neighborhood of λ such that the three sets I_0, I_P, I_N remain unchanged for the solution, equation (6) still hold so that we have

$$\beta_I(\lambda') = \beta_I(\lambda) - (\lambda' - \lambda) \left(X_I^\top X_I - \frac{1}{\gamma}\,\mathbf{diag}(\mathbf{r}_I) \right)^{-1} \begin{pmatrix} sign(\beta_P) \\ 0_N \end{pmatrix}.$$

The solution $\beta_I(\lambda)$ varies piecewise linearly with respect to λ. Note that since with MCP convexity is not guaranty, stationary point may be not unique and it may exists several local minima depending on the choice of the three sets I_0, I_P, I_N.

3.2 Subgradient Strategies

3.2.1 Subgradient iterations

Subgradient iterations consists in generating a sequence of points, using a sub gradient as descent direction. The subgradient of the MCP cost function is given in (4) with the gradient of the least square component $g = X^\top(X\beta - y)$. When $\beta_j = 0$, any subgradient can be taken. However, when $|g_j| \leq \lambda$, $\beta_j = 0$ fulfill for this component the optimality condition and therefore shouldn't be changed to promote sparsity in the solution. Otherwise,

```
I0 = find(abs(beta)<sqrt(eps));
 g = X'*(X*beta-y);
 g(I0)=(abs(g(I0)-beta(I0)/gam)-lambda).*(abs(g(I0) - beta(I0)/gam) > lambda);
 g = g+((lambda*sign(beta)-beta/gam).*(abs(beta)<lambda*gam)).*(abs(beta)>eps);
```

3.2.2 The component wise algorithm for MCP

The component wise algorithm consists in considering its univariate solution iteratively on all variables [Breheny and Huang, 2011, El Anbari, 2011]. The univariate solution in the one minimizing the MCP cost function with respect to a single component of vector β, say β_j, all the remaining one considered as constant. The univariate solution in the one canceling the sub differential of the MCP cost function with respect to parameter β_j, that is the set of all

$$\frac{\partial J_{\text{MCP}}(\beta)}{\partial \beta_j} = X_{\bullet j}^\top (X\beta - y) + \begin{cases} \lambda \alpha_j & \text{if } |\beta_j| = 0 \\ sign(\beta_j) \max\left(0, \lambda - \frac{|\beta_j|}{\gamma}\right) & \text{else,} \end{cases}$$

for $\alpha_j \in [-1, 1]$. Vector $X_{\bullet j}$ denotes the column j of matrix X. In that case

$$0 \in \frac{\partial J_{\text{MCP}}(\beta)}{\beta_j} \quad \Leftrightarrow \quad \beta_j = \begin{cases} 0 & \text{if } |(X_{\bullet j}^\top X_{\bullet j})^{-1} X_{\bullet j}^\top \mathbf{r}| \leq \lambda \\ s_j \left(|(X_{\bullet j}^\top X_{\bullet j})^{-1} X_{\bullet j}^\top \mathbf{r}| - \lambda \right) \dfrac{\gamma}{\gamma - 1} & \text{if } |(X_{\bullet j}^\top X_{\bullet j})^{-1} X_{\bullet j}^\top \mathbf{r}| \leq \lambda\gamma \\ (X_{\bullet j}^\top X_{\bullet j})^{-1} X_{\bullet j}^\top \mathbf{r} & \text{else,} \end{cases}$$

with s_j the sign of $(X_{\bullet j}^\top X_{\bullet j})^{-1} X_{\bullet j}^\top \mathbf{r}$ and $\mathbf{r} = X\beta - y - X_{\bullet j}\beta_j$ the residual error.

```
for i = 1:nbitemax
    ind = randperm(p);
    for j = 1:p
       grad = - (Xi(:,ind(j))'*(Xi*beta-yi - Xi(:,ind(j))*beta(ind(j))));
       beta(ind(j))=sign(grad)*max(0,min((abs(grad)-lam)/(1-1/gam),abs(grad)));
    end
end
```

3.2.3 The proximal of the MCP penalty

A way to deal with non differentiability is to use a proximal gradient algorithm by iteratively solving a proximal operator problem [Gong et al., 2013]. The proximal operator of the MCP penalty function is defined by

$$\mathbf{prox}_{\text{MCP}}(u) \;=\; \arg\min_{t \in \mathbb{R}} \; \tfrac{1}{2}\|t - u\|^2 + \mathbf{pen}_{\lambda,\gamma}(|t|) \; .$$

The solution of this minimization problem gives

$$\mathbf{prox}_{\text{MCP}}(u) \;=\; \begin{cases} 0 & \text{if } |u| \leq \lambda \\ sign(u)\,(|u| - \lambda)\,\dfrac{\gamma}{\gamma - 1} & \text{if } |u| \leq \lambda\gamma \\ u & \text{else.} \end{cases}$$

Given β^{old} next iterates is

$$\beta^{new} = \mathbf{prox}_{\text{MCP}}\big(\beta^{old} - \rho X^\top (X\beta - y)\big),$$

where $\rho > 0$ is a step size, constant or determined by line search.

```
 rho = 1/svds(Xi,1)^2;
 while max(abs(beta-b0)) > tol
     b0 = beta;
     g = X'*(X*beta-y);
     bg = beta - rho * g;
     s = min(abs(bg),(abs(bg)-lam*rho)*gam/(gam-pas));
     beta = sign(bg).*max(0,s);
end
```

3.2.4 Using the proximal of the L_1 norm

The proximal gradient algorithm can be used on a different splitting of the MCP cost function. Using equation (2), the MCP loss function can be also decomposed as the sum of a differentiable function $\phi(\beta) = \frac{1}{2}\|y - X\beta\|_2^2 - h(\beta)$ and a non differentiable one $\varphi(\beta) = \lambda\|\beta\|_1$. The idea is to apply the proximal projection os φ on the gradient of ϕ. The gradient of ϕ is (see the green curve on figure 1)

$$\nabla_\beta \phi(\beta) = X^\top(X\beta - y) - \begin{cases} sign(\beta_j)\lambda & \text{if } |\beta_j| > \lambda\gamma \\ \dfrac{\beta_j}{\gamma} & \text{else.} \end{cases}$$

The proximal operator of the L_1 penalty function φ is

$$\mathbf{prox}_\varphi(u) \;=\; \begin{cases} 0 & \text{if } |u| \le \lambda \\ sign(u)(|u| - \lambda) & \text{else.} \end{cases}$$

```
while max(abs(beta-b0)) > tol
    b0 = beta;
    g = Xi'*(Xi*beta-yi) - min(lam , abs(beta)/gam).*sign(beta);
    bg = beta - rho * g;
    beta = sign(bg).*max(0,abs(bg) - rho*lam);
end
```

3.3 The DC algorithm for MCP

This approach relies on difference of convex (DC) functions programming, that is to decompose the MCP cost function (2) into a difference of convex functions [Gasso et al., 2009]. Based on this decomposition, the DC algorithm amounts to iteratively building a sequence by minimizing, for a given β^{old}, the following convex surrogate cost function

$$J_{\text{DC}}(\beta) = \tfrac{1}{2}\|y - X\beta\|^2 + \lambda\|\beta\|_1 - \lambda \sum_{j=1}^{p} h'_{\lambda,\gamma}(|\beta_j^{old}|)\ |\beta_j|, \tag{7}$$

h' being the derivative of the Huber loss function. It turns out that in that case minimizing the DC cost function (7) can be seen as minimizing an adaptive Lasso [Zou, 2006]

$$\min_{\beta \in \mathbf{R}^p} \ \tfrac{1}{2}\|y - X\beta\|^2 + \lambda \sum_{j=1}^{p} w_j |\beta_j|, \tag{8}$$

with weights

$$w_j = \begin{cases} 0 & \text{if } |\beta_j^{old}| > \lambda\gamma \\ 1 - \dfrac{|\beta_j^{old}|}{\gamma\lambda} & \text{else} \end{cases} .$$

```
while max(abs(beta-b0)) > tol
    b0 = beta;
    w = max(0,lambda - abs(beta)/gam);
    [xnew, dual_var, pos] = monqp(H,c,[w ; w],b,inf,1,0); % Adaptive lasso
    Bpm = zeros(2*p,1);
    Bpm(pos) = xnew;
    beta = Bpm(1:p)-Bpm(p+1:end);
end
```

4 Experiments

4.1 Experimental setup

To evaluate the practical performance of the algorithms, we carry out some simulation studies. the linear model (1) has been used To generate the data with a gaussian noise of variance $\sigma = 1$. The design matrix X has been also randomly generated using a correlated Gaussian normalized distribution whose vector components were correlated with correlation between row j and k of matrix X equals r^{k-j} (as proposed by Zou [2006]). As usual, the design matrix X has been normalized column wise to mean zero and variance one. For all $p \geq 10$ the true value of the parameter was $\beta = (1, 2, 3, 4, 5, -1, -2, -3, -4, -5, 0, \dots, 0)^\top$ with ten non zero components completed with zeros. All methods were run until the same convergence criteria was met (*i.e.* where appropriate, that the step length between iterates, change in function value between iterates, negative directional derivative, or optimality condition was less than 10^{-6}.

4.2 Experimental results

- The convex case
  - DC may present unstable behavior depending on the solver used for the adaptive lasso.
  - Convergence criteria is an issue
- The non convex case
  - There may exists many stationary points. Breheny and Huang [2011] note that the solutions are continuous and stable in the locally convex regions, but may be discontinuous and erratic otherwise.
  - The solution provided by most of the algorithms strongly depends on the initial guess. Begin with low cost (least square with $\lambda = 0$), with a sparse solution ($\beta = 0$ and $\lambda = \infty$) or with the solution provided by the Lasso as advocated in Wang et al. [2014].
  - The homotopy method provides a different solution. It may take a sub optimal path (only in the non convex case)
- Large scale
  - Homotopy scales well with n
  - Homotopy hardly scale with p
  - For small λ prox's is two to ten times faster than CW.

Best one is prox L1

5 Discussion

For small p and large n the Homotopy algorithm is the one to be used. It provides good results for large λ and it is very fast. For small λ, the use of DC with the zero initialization may allow to get better results at the price of an important additional computational cost. For large p, the proximal algorithm provides a good compromise between accuracy and computation cost.

Acceleration - Convergence Improvement

Global convegence

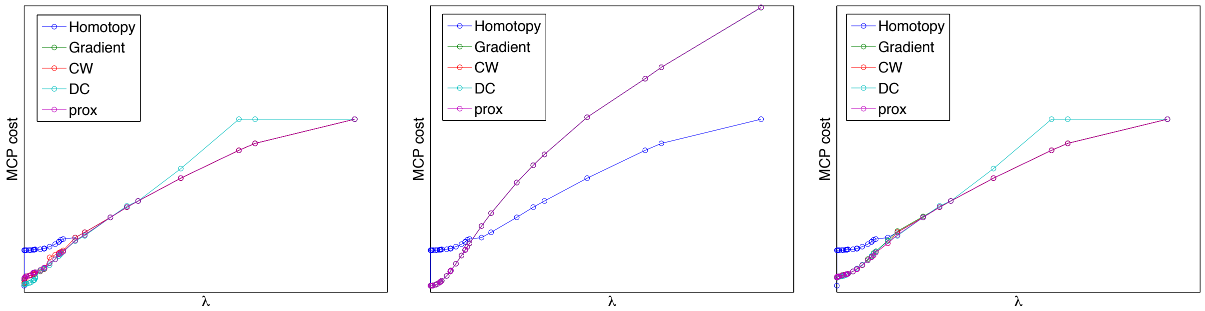

Figure 2: Illustration of the influence of the initialization. The MCP costs of the whole regularization paths of seven algorithms for $n = 20, p = 15, \gamma = 2.5$ (non convex case) when starting from zero (left), the least square solution (center) and the Lasso (right).

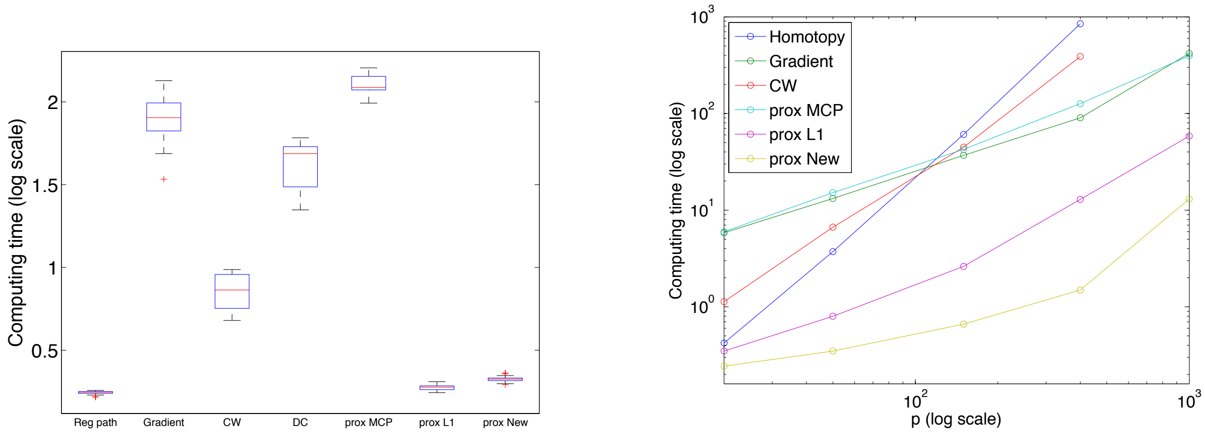

Figure 3: Left: the computing times of twenty replications of the whole regularization paths of seven algorithms for $n = 20000, p = 20, \gamma = 2$. Right: the computing time complexity curves as a function of p for $n = 20000, \gamma = 2$.

References

Adil Bagirov, Napsu Karmitsa, and Marko M Mäkelä. *Introduction to Nonsmooth Optimization: Theory, Practice and Software*. Springer, 2014.

Patrick Breheny and Jian Huang. Coordinate descent algorithms for nonconvex penalized regression, with applications to biological feature selection. *The annals of applied statistics*, 5 (1):232, 2011.

Frank H Clarke. *Optimization and nonsmooth analysis*, volume 5. SIAM, 1990.

Mohammed El Anbari. *Regularization and variable selection using penalized likelihood*. PhD thesis, Université de Paris-Sud 11 et Université Cadi Ayyad, 2011.

Gilles Gasso, Alain Rakotomamonjy, and Stéphane Canu. Recovering sparse signals with a certain family of nonconvex penalties and dc programming. *Signal Processing, IEEE Transactions on*, 57(12):4686–4698, 2009.

Pinghua Gong, Changshui Zhang, Zhaosong Lu, Jianhua Huang, and Jieping Ye. A general iterative shrinkage and thresholding algorithm for non-convex regularized optimization problems. In *Machine Learning, ICML*, pages 37–45, 2013.

Trevor Hastie, Robert Tibshirani, Jerome Friedman, T Hastie, J Friedman, and R Tibshirani. *The elements of statistical learning*, volume 2. Springer, 2009.

Zhaoran Wang, Han Liu, and Tong Zhang. Optimal computational and statistical rates of convergence for sparse nonconvex learning problems. *The Annals of Statistics*, 42(6):2164–2201, 12 2014.

Cun-Hui Zhang. Nearly unbiased variable selection under minimax concave penalty. *The Annals of Statistics*, pages 894–942, 2010.

Cun-Hui Zhang, Tong Zhang, et al. A general theory of concave regularization for high-dimensional sparse estimation problems. *Statistical Science*, 27(4):576–593, 2012.

Hui Zou. The adaptive lasso and its oracle properties. *Journal of the American statistical association*, 101(476):1418–1429, 2006.