Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

HHS Public Access

Author manuscript

Ann Stat. Author manuscript; available in PMC 2017 April 01.

Published in final edited form as:

Ann Stat. 2016 April ; 44(2): 629–659. doi:10.1214/15-AOS1380.

GLOBAL SOLUTIONS TO FOLDED CONCAVE PENALIZED NONCONVEX LEARNING

Hongcheng Liu[*], **Tao Yao**[*], and **Runze Li**[†]

The Pennsylvania State University

Abstract

This paper is concerned with solving nonconvex learning problems with folded concave penalty. Despite that their global solutions entail desirable statistical properties, there lack optimization techniques that guarantee global optimality in a general setting. In this paper, we show that a class of nonconvex learning problems are equivalent to general quadratic programs. This equivalence facilitates us in developing mixed integer linear programming reformulations, which admit finite algorithms that find a provably global optimal solution. We refer to this reformulation-based technique as the mixed integer programming-based global optimization (MIPGO). To our knowledge, this is the first global optimization scheme with a theoretical guarantee for folded concave penalized nonconvex learning with the SCAD penalty (Fan and Li, 2001) and the MCP penalty (Zhang, 2010). Numerical results indicate a significant outperformance of MIPGO over the state-of-the-art solution scheme, local linear approximation, and other alternative solution techniques in literature in terms of solution quality.

Keywords and phrases

Folded concave penalties; global optimization; high dimensional statistical learning; MCP; nonconvex quadratic programming; SCAD; sparse recovery

1. Introduction

Sparse recovery is of great interest in high-dimensional statistical learning. Among the most investigated sparse recovery techniques are LASSO and the nonconvex penalty methods, especially folded concave penalty techniques (see Fan, Xue and Zou, 2014, for a general definition). Although LASSO is a popular tool primarily because its global optimal solution is efficiently computable, recent theoretical and numerical studies reveal that this technique requires a critical irrepresentable condition to ensure statistical performance. In comparison,

[*]Support by Penn State Grace Woodward Collaborative Engineering/Medicine Research Grant, National Science Foundation grant CMMI 1300638, Marcus PSU-Technion Partnership grant and Mid-Atlantic University Transportation Centers grant.

[†]Support by National Science Foundation DMS 1512422 and National Institute of Health grants P50 DA036107 and P50 DA039838.

Department of Industrial and Manufacturing Engineering, Pennsylvania State University, University Park, PA 16802, USA, hql5143@psu.edu, taoyao@psu.edu

Department of Statistics, Pennsylvania State University, University Park, PA 16802, USA, rzli@psu.edu

SUPPLEMENTARY MATERIAL

Supplement to "Global solutions to folded concave penalized nonconvex learning":

(doi: COMPLETED BY THE TYPESETTER; .pdf). This supplemental material includes the proofs of Proposition 2.1, 2.3 and Lemma 4.1, and some additional numerical results.

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

the folded concave penalty methods require less theoretical regularity and entail better statistical properties (Zou, 2006; Meinshausen and Bühlmann, 2006; Fan, Xue and Zou, 2014). In particular, Zhang and Zhang (2012) showed that the global solutions to the folded concave penalized learning problems lead to a desirable recovery performance. However, these penalties cause the learning problems to be nonconvex and render the local solutions to be nonunique in general.

Current solution schemes in literature focus on solving a nonconvex learning problem locally. Fan and Li (2001) proposed a local quadratic approximation (LQA) method, which was further analyzed by using majorization minimization algorithm-based techniques in Hunter and Li (2005). Mazumder et al. (2011) and Breheny and Huang (2011) developed different versions of coordinate descent algorithms. Zou and Li (2008) proposed a local linear approximation (LLA) algorithm and Zhang (2010) proposed a PLUS algorithm. Kim, Choi, and Oh (2008) developed the ConCave Convex procedure (CCCP). To justify the use of local algorithms, conditions were imposed for the uniqueness of a local solution (Zhang, 2010; Zhang and Zhang, 2012); or, even if multiple local minima exist, the strong oracle property can be attained by LLA with wisely (but fairly efficiently) chosen initial solutions (Fan, Xue and Zou, 2014). Huang and Zhang (2012) showed that a multistage framework that subsumes the LLA can improve the solution quality stage by stage under some conditions. Wang, Kim and Li (2013) proved that calibrated CCCP produces a consistent solution path which contains the oracle estimator with probability approaching one. Loh and Wainwright (2015) established conditions for all local optima to lie within statistical precision of the true parameter vector, and proposed to employ the gradient method for composite objective function minimization by Nesterov (2007) to solve for one of the local solutions. Wang, Liu and Zhang (2014) incorporated the gradient method by Nesterov (2007) into a novel approximate regularization path following algorithm, which was shown to converge linearly to a solution with an oracle statistical property. Nonetheless, none of the above algorithms theoretically ensure global optimality.

In this paper, we seek to solve folded concave penalized nonconvex learning problems in a direct and generic way: to derive a reasonably efficient solution scheme with a provable guarantee on global optimality. Denote by n the sample size, and by d the problem dimension. Then the folded concave penalized learning problem of our discussion is formulated as following:

$$\min_{\beta\in\Lambda} \mathscr{L}(\beta) := \mathbb{L}(\beta) + n\sum_{i=1}^{d} P_\lambda(|\beta_i|), \quad (1.1)$$

where $P_\lambda(\cdot) : \mathbb{R} \to \mathbb{R}$ is a penalty function with tuning parameter λ. Our proposed procedure is directly applicable for settings allowing β_i to have different λ_i or different penalty. For ease of presentation and without loss of generality, we assume $P_\lambda(\cdot)$ is the same for all coefficients. Function $\mathbb{L}(\cdot) : \mathbb{R}^d \to \mathbb{R}$ is defined as a quadratic function,

$\mathbb{L}(\beta) := \frac{1}{2}\beta^\top Q\beta + q^\top\beta$, which is an abstract representation of a proper (quadratic) statistical loss function with $Q \in \mathbb{R}^{d\times d}$ and $q \in \mathbb{R}^d$ denoting matrices from data samples. Denote by $\Lambda := \{\beta \in \mathbb{R}^d : \mathcal{A}^\top\beta \le \mathbf{b}\}$ the feasible region defined by a set of linear constraints with $\mathcal{A} \in$

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

$\mathbb{R}^{d\times m}$ and $\mathbf{b} \in \mathbb{R}^m$ for some proper $m : 0 \le m < d$. Assume throughout the paper that Q is symmetric, $\mathcal{A}$ is full rank, and Λ is non-empty. Notice that under this assumption, the loss function does not have to be convex. We instead stipulate that problem (1.1) is well defined, i.e., there exists a finite global solution to (1.1). To ensure the well-definedness, it suffices to assume that the statistical loss function $\mathbb{L}(\beta)$ is bounded from below on Λ. As we will discussed in Section 2.1, penalized linear regression (least squares), penalized quantile regression, penalized linear support vector machine, penalized corrected linear regression and penalized semiparametric elliptical design regression can all be written in the unified form of (1.1). Thus, the problem setting in this paper is general enough to cover some new applications that are not addressed in Fan, Xue and Zou (2014). Specifically, the discussions in Fan, Xue and Zou (2014) covered sparse linear regression, sparse logistic regression, sparse precision matrix estimation, and sparse quantile regression. All these estimation problems intrinsically have convex loss functions. Wang, Liu and Zhang (2014) and Loh and Wainwright (2015) considered problems with less regularity by allowing the loss functions to be nonconvex. Their proposed approaches are, therefore, applicable to corrected linear regression and semiparametric elliptical design regression. Nonetheless, both works assumed different versions of restricted strong convexity. (See Section 4 for more discussions about restricted strong convexity.) In contrast, our analysis does not make assumptions of convexity, nor of any form of restricted strong convexity, on the statistical loss function. Moreover, the penalized support vector machine problem has been addressed in none of the above literature.

We assume $P_\lambda(\cdot)$ to be either one of the two mainstream folded concave penalties: (i) smoothly clipped absolute deviation (SCAD) penalty (Fan and Li, 2001), and (ii) minimax concave penalty (MCP, Zhang, 2010). Notice that both SCAD and MCP are nonconvex and nonsmooth. To facilitate our analysis and computation, we reformulate (1.1) into three well-known mathematical programs: firstly, a general quadratic program; secondly, a linear program with complementarity constraints; and finally, a mixed integer (linear) program (MIP). With these reformulations, we are able to formally state the worst-case complexity of computing a global optimum to folded concave penalized nonconvex learning problems. More importantly, with the MIP reformulation, the global optimal solution to folded concave penalized nonconvex learning problems can be numerically solved with a provable guarantee. This reformulation-based solution technique is referred to as the *MIP-based global optimization* (MIPGO).

In this paper, we make the following major contributions:

- **a.** We first establish a connection between folded concave penalized nonconvex learning and quadratic programming. This connection enables us to analyze the complexity of solving the problem globally.
- **b.** We provide an MIPGO scheme (namely, the MIP reformulations) to SCAD and MCP penalized nonconvex learning, and further prove that MIPGO ensures global optimality.

To our best knowledge, MIPGO probably is the first solution scheme that theoretically ascertains global optimality. In terms of both statistical learning and optimization, a global

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

optimization technique to folded concave penalized nonconvex learning is desirable. Zhang and Zhang (2012) provided a rigorous statement on the statistical properties of a global solution, while the existing solution techniques in literature cannot ensure a local minimal solution. Furthermore, the proposed MIP reformulation enables global optimization techniques to be applied directly to solving the original nonconvex learning problem instead of approximating with surrogate subproblems such as local linear or local quadratic approximations. Therefore, the objective of the MIP reformulation also measures the (in-sample) estimation quality. Due to the critical role of binary variables in mathematical programming, an MIP has been well studied in literature. Although an MIP is theoretically intractable, the computational and algorithmic advances in the last decade have made an MIP of larger problem scales fairly efficiently computable (Bertsimas et al., 2011). MIP solvers can further exploit the advances in computer architectures, e.g., algorithm parallelization, for additional computational power.

To test the proposed solution scheme, we conduct a series of numerical experiments comparing MIPGO with different existing approaches in literature. Involved in the comparison are a local optimization scheme (Loh and Wainwright, 2015), approximate path following algorithm (Wang, Liu and Zhang, 2014), LLA (Wang, Kim and Li, 2013; Fan, Xue and Zou, 2014), and two different versions of coordinate descent algorithms (Mazumder et al., 2011; Breheny and Huang, 2011). Our numerical results show that MIPGO can outperform all these alternative algorithms in terms of solution quality.

The rest of the paper is organized as follows. In Section 2, we introduce our setting, present some illustrative examples, and derive reformulations of nonconvex learning with the SCAD penalty and the MCP in the form of general quadratic programs. Section 3 formally states the complexity of approximating a global optimal solution and then derives MIPGO. Sections 4 and 5 numerically compare MIPGO with the techniques as perWang, Liu and Zhang (2014) and Loh and Wainwright (2015) and with LLA, respectively. Section 6 presents a more comprehensive numerical comparison with several existing local schemes. Section 7 concludes the paper. Some technical proofs are given in Section 8, and more technical proofs are given in the online supplement of this paper.

2. Setting, Example and Folded Concave Penalty Reformulation

It is worth noting that the abstract form (1.1) evidently subsumes a class of nonconvex learning problems with different statistical loss functions. Before we pursue further, let us provide a few examples of the loss functions that satisfy our assumptions to illustrate the generality of our statistical setting. Suppose that $\{(x_t, y_t) : t = 1, \ldots, n\} \subset \mathbb{R}^d \times \mathbb{R}$ is a random sample of size n. Let $y = (y_1, \ldots, y_n)^{\mathrm{T}}$ be the $n \times 1$ response vector, and $X = (x_1, \ldots, x_n)^{\mathrm{T}}$, the $n \times d$ design matrix. Denote throughout this paper by $\|\cdot\|_2$ the ℓ_2 norm and by $|\cdot|$ the ℓ_1 norm.

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

2.1. Examples

a. The ℓ_2 loss for the least squares problem, formulated as $\mathbb{L}_2(\beta) := \frac{1}{2}\sum_{t=1}^{n}(y_t - x_t^\top \beta)^2 = \frac{1}{2}\|y - X\beta\|_2^2$. It is easy to derive that the ℓ_2-loss can be written in the form of the loss function as in (1.1).

b. The ℓ_1 loss, formulated as $\mathbb{L}_1(\beta) := \sum_{t=1}^{n}|y_t - x_t^\top \beta| = |y - X\beta|$. In this case, we can instantiate the abstract form (1.1) into

$$\min_{\beta\in\mathbb{R}^d,\, \psi\in\mathbb{R}^n} \left\{ \mathbf{1}^\top \psi + n\sum_{i=1}^{d} P_\lambda(|\beta_i|) : -\psi \le y - X\beta \le \psi \right\},$$

where **1** denotes the all-ones vector with a proper dimension.

c. The quantile loss function in a quantile regression problem, defined as

$$\mathbb{L}_\tau(\beta) := \sum_{t=1}^{n}\rho_\tau(y_t - x_t^\top \beta) = \sum_{t=1}^{n}(y_t - x_t^\top \beta)\{\tau - \mathbb{I}(y_t < x_t^\top \beta)\},$$

where, for any given $\tau \in (0, 1)$, we have $\rho_t(u) := u\{\tau - \mathbb{I}(u < 0)\}$. This problem with a penalty term can be written in the form of (1.1) as

$$\min_{\beta\in\mathbb{R}^d,\, \psi\in\mathbb{R}^n} \mathbf{1}^\top[(y - X\beta)\tau + \psi] + n\sum_{i=1}^{d} P_\lambda(|\beta_i|), \quad s.t.\ \psi \ge X\beta - y; \quad \psi \ge 0.$$

d. The hinge loss function of a linear support vector machine classifier, which is formulated as $\mathbb{L}_{SVM} = \sum_{t=1}^{n}[1 - y_t x_t^\top \beta]_+$. Here, it is further assumed that $y_t \in \{-1, +1\}$, which is the class label, for all $t = 1, \ldots, n$. The corresponding instantiation of the abstract form (1.1) in this case can be written as

$$\min_{\beta\in\mathbb{R}^d,\, \psi\in\mathbb{R}^n} \mathbf{1}^\top \psi + n\sum_{i=1}^{d} P_\lambda(|\beta_i|), \quad s.t.\ \psi_t \ge 1 - y_t x_t^\top \beta; \quad \psi_t \ge 0 \quad \forall t = 1, \ldots, n.$$

e. Corrected linear regression and semiparametric elliptical design regression with a nonconvex penalty. According to Loh and Wainwright (2015) and Wang, Liu and Zhang (2014) both regression problems can be written as general quadratic functions and, therefore, they are special cases of (1.1) given that both problems are well defined.

2.2. Equivalence of Nonconvex Learning with Folded Concave Penalty to a General Quadratic Program

In this section, we provide equivalent reformulations of the nonconvex learning problems into a widely investigated form of mathematical programs, general quadratic programs. We will concentrate on two commonly-used penalties: the SCAD penalty and the MCP.

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

Specifically, given $a > 1$ and $\lambda > 0$, the SCAD penalty (Fan and Li, 2001) is defined as:

$$P_{SCAD,\lambda}(\theta) := \int_0^\theta \lambda \left\{ \mathbb{I}(t \le \lambda) + \frac{(a\lambda - t)_+}{(a-1)\lambda} \mathbb{I}(t > \lambda) \right\} dt, \quad (2.1)$$

where $\mathbb{I}(\cdot)$ is the indicator function, and $(b)_+$ denotes the positive part of b.

Given $a > 0$ and $\lambda > 0$, the MCP (Zhang, 2010) is defined as:

$$P_{MCP,\lambda}(\theta) := \int_0^\theta \frac{(a\lambda - t)_+}{a} dt. \quad (2.2)$$

We first provide the reformulation of (1.1) with SCAD penalty to a general quadratic program in Proposition 2.1, whose proof will be given in the online supplement. Let $\mathcal{F}_{SCAD}(\cdot, \cdot) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ be defined as

$$\mathscr{F}_{SCAD}(\beta, g) = \frac{1}{2}\beta^\top Q\beta + q^\top \beta + n\sum_{i=1}^d \left\{ (|\beta_i| - a\lambda) \cdot g_i + \frac{1}{2}(a-1) \cdot g_i^2 \right\},$$

where $\beta = (\beta_i) \in \mathbb{R}^d$ and $g = (g_i) \in \mathbb{R}^d$.

Proposition 2.1: *Let* $P_\lambda(\cdot) = P_{SCAD,\lambda}(\cdot)$.

a. *The minimization problem* (1.1) *is equivalent to the following program*

$$\min_{\beta \in \Lambda,\, g \in [0,\lambda]^d} \mathscr{F}_{SCAD}(\beta, g). \quad (2.3)$$

b. *The first derivative of the SCAD penalty can be rewritten as*

$$P'_\lambda(\theta) = \operatorname{argmin}_{\kappa \in [0,\lambda]} \left\{ (\theta - a\lambda) \cdot \kappa + \tfrac{1}{2}(a-1) \cdot \kappa^2 \right\} \text{ for any } \theta \ge 0.$$

To further simplify the formulation, we next show that program (2.3), as an immediate result of Proposition 2.1, is equivalent to a general quadratic program.

Corollary 2.2: *Program* (2.3) *is equivalent to*

$$\min_{\beta,g,h \in \mathbb{R}^d} \frac{1}{2}(\beta^\top Q\beta + n(a-1)g^\top g + 2ng^\top h) + q^\top \beta - na\lambda \mathbf{1}^\top g \quad s.t.\ \beta \in \Lambda; \quad h \ge \beta; \quad h \ge -\beta; \quad 0 \le g \le \lambda. \quad (2.4)$$

Proof: The proof is completed by invoking Proposition 2.1 and the non-negativity of g.

The above reformulation facilitates our analysis by connecting the non-convex learning problem with a general quadratic program. The latter has been heavily investigated in literature. Interested readers are referred to Vavasis (1991) for an excellent summary on computational issues in solving a nonconvex quadratic program.

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

Following the same argument for the SCAD penalty, we have similar findings for (1.1) with the MCP. The reformulation of (1.1) with the MCP is given in the following proposition, whose proof will be given in the online supplement. Let $\mathcal{F}_{MCP}(\cdot,\cdot):\mathbb{R}^d\times\mathbb{R}^d\to\mathbb{R}$ be defined as

$$\mathscr{F}_{MCP}(\beta,g):=\frac{1}{2}\beta^\top Q\beta+q^\top\beta+n\sum_{i=1}^{d}\left\{\frac{1}{2a}g_i^2-\left(\frac{1}{a}g_i-\lambda\right)|\beta_i|\right\}.$$

Proposition 2.3: *Let* $P_\lambda(\cdot)=P_{MCP,\lambda}(\cdot)$.

a. *The model* (1.1) *is equivalent to the following program*

$$\min_{\beta\in\Lambda,\, g\in[0,\, a\,\lambda]^d}\mathscr{F}_{MCP}(\beta,g). \quad (2.5)$$

b. *For any* $\theta\geq 0$*, the first derivative of the MCP can be rewritten as* $P'_\lambda(\theta)=\frac{a\lambda-g^*(\theta)}{a}$ *where* $g^*(\cdot):\mathbb{R}_+\to\mathbb{R}$ *is defined as*

$$g^*(\theta):=\operatorname*{argmin}_{\kappa\in[0,a\lambda]}\left\{\frac{1}{2a}\kappa^2-\left(\frac{1}{a}\kappa-\lambda\right)\theta\right\}.$$

Immediately from the above theorem is an equivalence between MCP penalized nonconvex learning and the following nonconvex quadratic program.

Corollary 2.4: *The program* (2.5) *is equivalent to*

$$\min_{\beta,g,h\in\mathbb{R}^d}\frac{1}{2}\left(\beta^\top Q\beta+\frac{n}{a}g^\top g-\frac{2n}{a}g^\top h\right)+n\lambda\mathbf{1}^\top h+q^\top\beta\quad s.t.\ \beta\in\Lambda;\quad h\geq\beta;\quad h\geq-\beta;\quad 0\leq g\leq a\lambda. \quad (2.6)$$

Proof: This is a direct result of Proposition 2.3 by noting the non-negativity of g.

With the above reformulations we are able to provide our complexity analysis and devise our promised solution scheme.

3. Global Optimization Techniques

This section is concerned with global optimization of (1.1) with the SCAD penalty and the MCP. We will first establish the complexity of approximating an ε-suboptimal solution in Section 3.1 and then provide the promised MIPGO method in Section 3.2. Note that, since the proposed reformulation di3erentiates between solving nonconvex learning with the SCAD penalty and solving nonconvex learning with the MCP, we will use MIPGO-SCAD or MIPGO-MCP to rule out the possible ambiguity occasionally.

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

3.1. Complexity of Globally Solving Folded Concave Penalized Nonconvex Learning

In Section 2, we have shown the equivalence between (1.1) and a quadratic program in both the SCAD and MCP cases. Such equivalence allows us to immediately apply existing results for quadratic programs to the complexity analysis of (1.1). We first introduce the concept of ε-approximate of global optimum that will be used Theorem 3.1(c). Assume that (1.1) has finite global optimal solutions. Denote by $\beta^* \in \Lambda$ a finite, globally optimal solution to (1.1). Following Vavasis (1992), we call β_ε^* to be an ε-approximate solution if there exists another feasible solution $\bar{\beta} \in \Lambda$ such that

$$\mathscr{L}(\beta_\varepsilon^*) - \mathscr{L}(\beta^*) \leq \varepsilon[\mathscr{L}(\bar{\beta}) - \mathscr{L}(\beta^*)] \quad (3.1)$$

Theorem 3.1

a. *Denote by* $I_{d\times d} \in \mathbb{R}^{d\times d}$ *an identity matrix, and by*

$$\mathscr{H}_1 := \begin{bmatrix} Q & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & n(a-1)I_{d\times d} & nI_{d\times d} \\ \mathbf{0} & nI_{d\times d} & \mathbf{0} \end{bmatrix} \quad (3.2)$$

the Hessian matrix of (2.4)*. Let* $1 < a < \infty$*, then* $\mathcal{H}_1$ *has at least one negative eigenvalue (i.e.,* $\mathcal{H}_1$ *is not positive semidefinite).*

b. *Denote by*

$$\mathscr{H}_2 := \begin{bmatrix} Q & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & n/aI_{d\times d} & -n/aI_{d\times d} \\ \mathbf{0} & -n/aI_{d\times d} & \mathbf{0} \end{bmatrix}. \quad (3.3)$$

the Hessian matrix of (2.6)*. Let* $0 < a < \infty$*, then* $\mathcal{H}_2$ *has at least one negative eigenvalue.*

c. *Assume that* (1.1) *has finite global optimal solutions. Problem* (1.1) *admits an algorithm with complexity of* $O(\lceil 3d(3d+1)/\sqrt{\varepsilon}\rceil^r l)$ *to attain an* ε*-approximate of global optimum, where* l *denotes the worst-case complexity of solving a convex quadratic program with* $3d$ *variables, and* r *is the number of negative eigenvalues of* $\mathcal{H}_1$ *for the SCAD penalty, and the number of negative eigenvalues of* $\mathcal{H}_2$ *for the MCP.*

Proof: Consider an arbitrary symmetric matrix Θ. Throughout this proof, $\Theta \succeq 0$ means that Θ is positive semidefinite.

Notice $\mathcal{H}_1 \succeq 0$ only if

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

$$\mathscr{B}_1 := \begin{bmatrix} n(a-1)I_{d\times d} & nI_{d\times d} \\ nI_{d\times d} & \mathbf{0} \end{bmatrix} \succcurlyeq 0. \quad (3.4)$$

Since $1 < a < \infty$, we have $n(a-1)I_{d\times d} \succ 0$. By Schur complement condition, the positive semidefiniteness of $\mathcal{B}_1$ requires that $-n(a-1)^{-1} \geq 0$, which contradicts with the assumption $1 < a < \infty$. Therefore, $\mathcal{H}_1$ is not positive semidefinite. This completes the proof of (a).

In order to show (b), similarly, we have $\mathcal{H}_2 \succeq 0$ only if

$$\mathscr{B}_2 := \begin{bmatrix} n/aI_{d\times d} & -n/aI_{d\times d} \\ -n/aI_{d\times d} & \mathbf{0} \end{bmatrix} \succcurlyeq 0 \quad (3.5)$$

Since $0 < a < \infty$, we have $n/aI_{d\times d} \succ 0$. By Schur complement condition, the positive semidefiniteness of $\mathcal{B}_2$ requires that $-n/a \geq 0$, which contradicts with the assumption $0 < a < \infty$. Therefore, $\mathcal{H}_2$ is not positive semidefinite, which means $\mathcal{H}_2$ has at least one negative eigenvalue. This completes the proof for part (b).

Part (c) can be shown immediately from Theorem 2 in Vavasis (1992), and from the equivalence between (1.1) and the quadratic program (2.4) for the SCAD case, and that between (1.1) and (2.6) for the MCP case.

In Theorem 3.1(c), the complexity result for attaining such a solution is shown in an abstract manner and no practically implementable algorithm has been proposed to solve a nonconvex quadratic program in general, or to solve (2.4) or (2.6) in particular.

Pardalos (1991) provided an example for a nonconvex quadratic program with 2^r local solutions. Therefore, by the equivalence between (2.4) (or (2.6)) for SCAD (or MCP) and (1.1), the latter may also have 2^r local solutions in some bad (not necessarily the worst) cases.

3.2. Mixed Integer Programming-based Global Optimization Technique

Now we are ready to provide the proposed MIPGO, which essentially is a reformulation of nonconvex learning with the SCAD penalty or the MCP into an MIP problem. Our reformulation is inspired by Vandenbussche and Nemhauser (2005), who provided MIP to a quadratic program with box constraints.

It is well known that an MIP can be solved with provable global optimality by solution schemes such as the branch-and-bound algorithm (B&B, Martí and Reinelt, 2011). Essentially, the B&B algorithm keeps track of both a global lower bound and a global upper bound on the objective value of the global minimum. These bounds are updated by B&B by systematically partitioning the feasible region into multiple convex subsets and evaluating the feasible and relaxed solutions within each of the partitions. B&B then refines partitions repetitively over iterations. Theoretically, the global optimal solution is achieved, once the gap between the two bounds is zero. In practice, the B&B is terminated until the two bounds

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

are close enough. The state-of-the-art MIP solvers incorporate B&B with additional features such as local optimization and heuristics to facilitate computation.

3.2.1. MIPGO for Nonconvex Learning with the SCAD Penalty—Let us introduce a notation. For two d-dimensional vectors $\Phi = (\phi_i) \in \mathbb{R}^d$ and $\Delta = (\delta_i) \in \mathbb{R}^d$, a complementarity constraint $0 \le \Phi \perp \Delta \ge 0$ means that $\phi_i \ge 0$, $\delta_i \ge 0$, and $\phi_i \delta_i = 0$ for all $i: 1 \le i \le d$. A natural representation of this complementarity constraint is a set of logical constraints involving binary variables $\mathbf{z} = (z_i) \in \{0, 1\}^d$:

$$\Phi \ge 0; \Delta \ge 0; \Phi \le \mathscr{M}\mathbf{z}; \Delta \le \mathscr{M}(1-\mathbf{z}); \mathbf{z} \in \{0,1\}^d. \quad (3.6)$$

The following theorem gives the key reformulation that will lead to the MIP reformulation.

Theorem 3.2: *Program* (2.4) *is equivalent to a linear program with (linear) complementarity constraints (LPCC) of the following form:*

$$\min \frac{1}{2}q^\top\beta - \frac{1}{2}\mathbf{b}^\top\rho - \frac{1}{2}na\lambda\mathbf{1}^\top g - \frac{1}{2}\lambda\gamma_4^\top\mathbf{1}; \quad s.t. \quad (3.7)$$

$$\begin{cases} Q\beta + q + \gamma_1 - \gamma_2 + \mathscr{A}\rho = 0; \quad ng - \gamma_1 - \gamma_2 = 0 \\ n(a-1)g + nh - na\lambda\mathbf{1} - \gamma_3 + \gamma_4 = 0 \\ 0 \le \gamma_1 \top h - \beta \ge 0; \quad 0 \le \gamma_2 \top h + \beta \ge 0 \\ 0 \le \gamma_3 \top g \ge 0; \quad 0 \le \gamma_4 \top \lambda - g \ge 0 \\ 0 \le \rho \top \mathbf{b} - \mathscr{A}^\top\beta \ge 0 \\ \beta, g, h, \gamma_1, \gamma_2, \gamma_3, \gamma_4 \in \mathbb{R}^d; \quad \rho \in \mathbb{R}^m. \end{cases} \quad (3.8)$$

The above LPCC can be immediately rewritten into an MIP. Rewriting the complementarity constraints in (3.8) into the system of logical constraints following (3.6), problem (2.4) now becomes

$$\min \frac{1}{2}q^\top\beta - \frac{1}{2}\mathbf{b}^\top\rho - \frac{1}{2}na\lambda\mathbf{1}^\top g - \frac{1}{2}\lambda\gamma_4^\top\mathbf{1}; \quad s.t. \quad (3.9)$$

$$\begin{cases} Q\beta + q + \gamma_1 - \gamma_2 + \mathscr{A}\rho = 0; \quad ng - \gamma_1 - \gamma_2 = 0 \\ n(a-1)g + nh - na\lambda\mathbf{1} - \gamma_3 + \gamma_4 = 0 \\ \gamma_1 \le \mathscr{M}\mathbf{z}_1;\ h_i - \beta_i \le \mathscr{M}(1-\mathbf{z}_1) \\ \gamma_2 \le \mathscr{M}\mathbf{z}_2;\ h + \beta \le \mathscr{M}(1-\mathbf{z}_2) \\ \gamma_3 \le \mathscr{M}\mathbf{z}_3;\ g_i \le \mathscr{M}(1-\mathbf{z}_3) \\ \gamma_4 \le \mathscr{M}\mathbf{z}_4;\ \lambda - g \le \mathscr{M}(1-\mathbf{z}_4) \\ \rho \le \mathscr{M}\mathbf{z}_5;\ \mathbf{b} - \mathscr{A}^\top\beta \le \mathscr{M}(1-\mathbf{z}_5) \\ -\beta \le h; \quad \gamma_2 \ge 0; \quad \beta \le h; \quad \gamma_1 \ge 0 \\ g \ge 0; \quad \gamma_3 \ge 0; \quad g \le \lambda; \quad \gamma_4 \ge 0 \\ \rho \ge 0;\ \mathbf{b} - \mathscr{A}^\top\beta \ge 0 \\ \mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3, \mathbf{z}_4 \in \{0,1\}^d;\ \mathbf{z}_5 \in \{0,1\}^m \\ \beta, g, h, \gamma_1, \gamma_2, \gamma_3, \gamma_4 \in \mathbb{R}^d; \quad \rho \in \mathbb{R}^m, \end{cases} \quad (3.10)$$

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

where we recall that $\mathcal{M}$ is a properly large constant.

The above program is in the form of an MIP, which admits finite algorithms that ascertain global optimality.

Theorem 3.3: *Program* (3.9)–(3.10) *admits algorithms that attain a global optimum in finite iterations.*

Proof: The problem can be solved globally in finite iterations by B&B (Lawler and Wood, 1966) method.

The proof in fact provides a class of numerical schemes that solve (3.9)–(3.10) globally and finitely. Some of these schemes have become highly developed and even commercialized. We elect to solve the above problem using one of the state-of-the-art MIP solvers, Gurobi, which is a B&B-based solution tool. (Detailed information about Gurobi can be found at http://www.gurobi.com/.)

3.2.2. MIPGO for Nonconvex Learning with the MCP—Following almost the same argument for the SCAD penalized nonconvex learning, we can derive the reformulation of the MCP penalized nonconvex learning problem into an LPCC per the following theorem.

Theorem 3.4: *Program* (2.6) *is equivalent to the following LPCC:*

$$\min \frac{1}{2}q^{\top}\beta-\frac{1}{2}\mathbf{b}^{\top}\rho-\frac{1}{2}a\lambda\mathbf{1}^{\top}\eta_4-\frac{1}{2}\lambda n\mathbf{1}^{\top}h; \quad s.t. \tag{3.11}$$

$$\begin{cases} Q\beta+q+\eta_1-\eta_2+\mathscr{A}\rho=0 \\ \frac{n}{a}g-\frac{n}{a}h-\eta_3+\eta_4=0; \quad -n(\frac{1}{a}g+\lambda\mathbf{1})-\eta_1-\eta_2=0 \\ 0\leq\eta_1\top h-\beta\geq 0; \quad 0\leq\eta_2\top h+\beta\geq 0 \\ 0\leq\eta_3\top g\geq 0; \quad 0\leq\eta_4\top a\lambda\mathbf{1}-g\geq 0 \\ 0\leq\rho\top\mathbf{b}-\mathscr{A}^{\top}\beta\geq 0 \\ \beta,\, g,\, h,\, \eta_1,\, \eta_2,\, \eta_3,\, \eta_4\in\mathbb{R}^d; \quad \rho\in\mathbb{R}^m \end{cases} \tag{3.12}$$

To further facilitate the computation, Program (3.11)–(3.12) can be represented as

$$\min \frac{1}{2}q^{\top}\beta-\frac{1}{2}\mathbf{b}^{\top}\rho-\frac{1}{2}a\lambda\mathbf{1}^{\top}\eta_4-\frac{1}{2}\lambda n\mathbf{1}^{\top}h; \quad s.t. \tag{3.13}$$

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

$$\begin{cases} q+Q\beta+\eta_1-\eta_2+\mathscr{A}\rho=0;\ \frac{n}{a}g-\frac{n}{a}h-\eta_3+\eta_4=0 \\ -n(\frac{1}{a}g+\lambda\mathbf{1})-\eta_1-\eta_2=0 \\ 0\le\eta_1\le\mathscr{M}\mathbf{z}_1;\ 0\le h-\beta\le\mathscr{M}(1-\mathbf{z}_1) \\ 0\le\eta_2\le\mathscr{M}\mathbf{z}_2;\ 0\le h+\beta\le\mathscr{M}(1-\mathbf{z}_2) \\ 0\le\eta_3\le M\mathbf{z}_3;\ 0\le g\le\mathscr{M}(1-\mathbf{z}_3) \\ 0\le\eta_4\le\mathscr{M}\mathbf{z}_4;\ 0\le a\lambda\mathbf{1}-g\le\mathscr{M}(1-\mathbf{z}_4) \\ 0\le\rho\le\mathscr{M}\mathbf{z}_5;\ \mathbf{b}-\mathscr{A}^\top\beta\le\mathscr{M}(1-\mathbf{z}_5);\ \mathbf{b}-\mathscr{A}^\top\beta\ge 0 \\ \mathbf{z}_1,\ \mathbf{z}_2,\ \mathbf{z}_3,\ \mathbf{z}_4\in\{0,\ 1\}^d;\ \mathbf{z}_5\in\{0,\ 1\}^m \\ \eta_1,\ \eta_2,\ \eta_3,\ \eta_4\in\mathbb{R}^d;\ \rho\in\mathbb{R}^m \end{cases} \tag{3.14}$$

The computability of a global optimal solution to the above MIP is guaranteed by the following theorem.

Theorem 3.5: *Program* (3.13)–(3.14) *admits algorithms that attain a global optimum in finite iterations.*

Proof: The problem can be solved globally in finite iterations by B&B (Lawler and Wood, 1966) method.

Combining the reformulations in Section 3.2, we want to remark that the MIP reformulation connects the SCAD or MCP penalized nonconvex learning with the state-of-the-art numerical solvers for MIP. This reformulation guarantees global minimum theoretically and yields reasonable computational expense in solving (1.1). To acquire such a guarantee, we do not impose very restrictive conditions. To our knowledge, there is no existing global optimization technique for the nonconvex learning with the SCAD penalty or the MCP penalty in literature under the same or less restrictive assumptions. More specifically, for MIPGO, the only requirement on the statistical loss function is that it should be a lower-bounded quadratic function on the feasible region Λ with the Hessian matrix Q being symmetric. As we have mentioned in Section 2, an important class of sparse learning problems naturally satisfy our assumption. In contrast, LLA, per its equivalence to a majorization minimization algorithm, converges asymptotically to a stationary point that does not differentiate among local maxima, local minima, or saddle points. Hence, the resulting solution quality is not generally guaranteed. Fan, Xue and Zou (2014) proposed the state-of-the-art LLA variant. It requires restricted eigenvalue conditions to ensure convergence to an oracle solution in two iterations with a lower-bounded probability. The convergence of the local optimization algorithms by Loh and Wainwright (2015) and Wang, Liu and Zhang (2014) both require the satisfaction of (conditions that imply) RSC. To our knowledge, MIPGO stipulates weaker conditions in contrast to the above solution schemes.

3.2.3. Numerical Stability of MIPGO—The representations of SCAD or MCP penalized nonconvex learning problems as MIPs introduce dummy variables to the original problem. These dummy variables are in fact Lagrangian multipliers in the KKT conditions of (2.4) or (2.6). In cases when no finite Lagrangian multipliers exist, the proposed MIPGO can result in numerical instability. To address this issue, we study an abstract form of SCAD or MCP penalized the nonconvex learning problems given as following.

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

$$\min\left\{\mathscr{F}(\beta,\, h,\, g)\colon (\beta,\, h)\in\tilde{\Lambda},\, g\in[0,\, M]^d\right\}, \quad (3.15)$$

where $M>0$, $\tilde{\Lambda}:=\{(\beta, h)\colon \beta\in\Lambda,\, h\in\mathbb{R}^d,\, h\geq\beta,\, h\geq-\beta\}$, and $\mathscr{F}:\tilde{\Lambda}\times[0, M]^d\to\mathbb{R}$ is assumed continuously differentiable in (β, h, g) with the gradient $\nabla\mathscr{F}$ being Lipschitz continuous. It may easily be verified that (2.4) and (2.6) are both special cases of (3.15). Now, we can write out the KKT conditions of this abstract problem as:

$$\nabla_\beta\mathscr{F}(\beta,\, h,\, g)+v_1-v_2+\mathscr{A}\rho=0 \quad (3.16)$$

$$\nabla_h\mathscr{F}(\beta,\, h,\, g)-v_1-v_2=0 \quad (3.17)$$

$$\nabla_g\mathscr{F}(\beta,\, g)-\zeta_1+\zeta_2=0 \quad (3.18)$$

$$\begin{array}{l}\zeta_{1,i}\cdot g_i=0;\quad \zeta_{2,i}\cdot(g_i-M)=0\\ \zeta_{1,i},\, \zeta_{2,i}\geq 0\end{array}\qquad \forall i=1,\ldots,d \quad (3.19)$$

$$\begin{array}{l}v_{1,i}\cdot(\beta_i-h_i)=0;\quad v_{2,i}\cdot(-\beta_i-h_i)=0\\ v_{1,i}\geq 0;\quad v_{2,i}\geq 0\end{array}\qquad \forall i=1,\ldots,d \quad (3.20)$$

$$\rho\geq 0;\quad \rho^\top(\mathscr{A}^\top\beta-\mathbf{b})=0, \quad (3.21)$$

where $\nabla_\beta\mathscr{F}(\beta, h, g):=\partial\mathscr{F}(\beta, h, g)/\partial\beta$, $\nabla_g\mathscr{F}(\beta, h, g):=\partial\mathscr{F}(\beta, h, g)/\partial g$, and $\nabla_h\mathscr{F}(\beta, h, g):=\partial\mathscr{F}(\beta, h, g)/\partial h$, and where $\zeta_1, \zeta_2, v_1, v_2\in\mathbb{R}^d$ and $\rho\in\mathbb{R}^m$ are the Lagrangian multipliers that we are concerned with. For convenience, the i-th dimension ($i=\{1, \ldots, d\}$) of these multipliers are denoted as $\zeta_{1,i}, \zeta_{2,i}, v_{1,i}, v_{2,i}$ and ρ_i, respectively. Notice that, since $\mathscr{A}$ is full-rank, then ρ is bounded if $\|\mathscr{A}\rho\|$ is bounded, where we let $\|\cdot\|$ be an ℓ_p norm with arbitrary $1\leq p\leq\infty$. (To see this, observe that $\|\mathscr{A}\rho\|_2=\sqrt{\rho^\top\mathscr{A}^\top\mathscr{A}\rho}$ and $\mathscr{A}^\top\mathscr{A}$ is positive definite.)

Theorem 3.6: *Denote a global optimal solution to problem* (3.15) *as* (β^*, h^*, g^*). *Assume that there exists a positive constant* C_1 *such that*

$$\max\{\|\nabla_\beta\mathscr{F}(\beta^*, h^*, g^*)\|,\, \|\nabla_h\mathscr{F}(\beta^*, h^*, g^*)\|,\, \|\nabla_g\mathscr{F}(\beta^*, h^*, g^*)\|\}\leq C_1.$$

Then the Lagrangian multipliers corresponding to this global optimum, $v_1, v_2, \zeta_1, \zeta_2$, *and* ρ *satisfy that*

$$\max\{\|v_1\|,\, \|v_2\|,\, \|\zeta_1\|,\, \|\zeta_2\|\}\leq C_1;\quad \textit{and}\quad \|\mathscr{A}\rho\|\leq 3C_1. \quad (3.22)$$

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

Proof: Recall that Λ is non-empty. Since $\mathcal{A}$ is full rank and all other constraints are linear and non-degenerate, we have the linear independence constraint qualification satisfied at a global solution, which then satisfies the KKT condition. (i) In order to show that υ_1 and υ_2 are bounded, with (3.17), we have $\|\upsilon_1 + \upsilon_2\| = \|\nabla_h \mathcal{F}(\beta^*, h^*, g^*)\| \le C_1$. Noticing the non-negativity of υ_1 and υ_2, we obtain $\max\{\|\upsilon_1\|, \|\upsilon_2\|\} \le \|\upsilon_1 + \upsilon_2\| = \|\nabla_h \mathcal{F}(\beta^*, h^*, g^*)\| \le C_1$. (ii) To show $\mathcal{A}\rho$ is bounded, considering (3.16), $\|\mathcal{A}\rho\| = \|\nabla_\beta \mathcal{F}(\beta, h, g) + \upsilon_1 - \upsilon_2\| \le \|\nabla_h \mathcal{F}(\beta^*, h^*, g^*)\| + \|\upsilon_1\| + \|\upsilon_2\| \le 3C_1$. (iii) To show that ζ_1 and ζ_2 are bounded, we notice that, immediately from (3.19), $\zeta_{1,i} \ge 0$; $\zeta_{2,i} \ge 0$; and $\zeta_{1,i} \cdot \zeta_{2,i} = 0$, for all $i = 1, \ldots, d$. Thus, according to (3.18), $C_1 \ge \|\nabla_g \mathcal{F}(\beta^*, h^*, g^*)\| \ge \|(\max\{\zeta_{1,i}, \zeta_{2,i}\}, i = 1, \ldots, d)\|$. Therefore, $\|\zeta_1\| \le C_1$ and $\|\zeta_2\| \le C_1$.

With Theorem 3.6, we claim that the Lagrangian multipliers corresponding to a global optimal solution cannot be arbitrarily large under proper assumptions. Hence, we conclude that the proposed method can be numerically stable. In practice, because ∇F is assumed Lipschitz continuous, we can simply impose an additional constraint $\|\beta\|_\infty \le C$ in the MIP reformulation for some positive constant C to ensure the satisfaction of (3.22). Conceivably, this additional constraint does not result in a significant modification to the original problem.

4. Comparison with the Gradient Methods

This section will compare MIPGO with Loh and Wainwright (2015) and Wang, Liu and Zhang (2014) when $\mathbb{L} := \mathbb{L}_2$. Thus the complete formulation is given as:

$$\min_{\beta \in \mathbb{R}^d} \mathcal{L}(\beta) = \frac{1}{2}\|y - X\beta\|_2^2 + n\sum_{i=1}^{d} P_\lambda(|\beta_i|). \quad (4.1)$$

where X and y are defined as in Section 2.1. We will refer to this problem as SCAD (or MCP) penalized linear regression (LR-SCAD (or -MCP)). To solve this problem, Loh and Wainwright (2015) and Wang, Liu and Zhang (2014) independently developed two types of computing procedures based on the gradient method proposed by Nesterov (2007). For the sake of simplicity, we will refer to both approaches as the gradient methods hereafter, although they both present substantial differentiation from the original gradient algorithm proposed by Nesterov (2007). To ensure high computational and statistical performance, both Loh and Wainwright (2015) and Wang, Liu and Zhang (2014) considered conditions called “restricted strong convexity” (RSC). We will illustrate in this section that RSC can be a fairly strong condition in LR-SCAD or -MCP problems and that MIPGO may potentially outperform the gradient methods regardless of whether the RSC is satisfied.

In Loh and Wainwright (2015) and Wang, Liu and Zhang (2014), RSC is defined differently. These two versions of definitions are discussed as below: Let $\beta_{true} = (\beta_{true,i}) \in \mathbb{R}^d$ be the true parameter vector and $k = \|\beta_{true}\|_0$. Denote that $L(\beta) := \frac{1}{2n}\|y - X\beta\|_2^2$. Then according to Loh and Wainwright (2015), $L(\beta)$ is said to satisfy RSC if the following inequality holds:

$$L(\beta') - L(\beta'') - \langle \nabla_\beta L(\beta''), \beta' - \beta'' \rangle \ge \begin{cases} \alpha_1 \|\beta' - \beta''\|_2^2 - \tau_1 \frac{\log d}{n} |\beta' - \beta''|^2 & \text{for all } \|\beta' - \beta''\|_2 \le 3 \\ \alpha_2 \|\beta' - \beta''\|_2 - \tau_2 \frac{\log d}{n} |\beta' - \beta''| & \text{for all} \|\beta' - \beta''\|_2 \ge 3 \end{cases} \quad (4.2)$$

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

for some a_1, $a_2 > 0$ and τ_1, $\tau_2 \geq 0$. Furthermore, Loh and Wainwright (2015) assumed (in Lemma 3 of their paper) that $\frac{64k\tau\log d}{n}+\mu \leq \alpha$ with $a = \min\{a_1, a_2\}$ and $\tau = \max\{\tau_1, \tau_2\}$, for some $\mu \geq 0$ such that $\mu\|\beta\|_2^2+\sum_{i=1}^{p} P_\lambda(|\beta_i|)$ is convex.

Wang, Liu and Zhang (2014) discussed a different version of RSC. They reformulated (4.1) into $\frac{\mathscr{L}(\beta)}{n}=\tilde{L}(\beta)+\lambda|\beta|$, where $\tilde{L}(\beta):=L(\beta)+\sum_{i=1}^{d} P_\lambda(\beta_i)-\lambda|\beta|$. According to the same paper, one can quickly check that $L(\tilde{\beta})$ is continuously differentiable. Then, their version of RSC, as in Lemma 5.1 of their paper, is given as:

$$\tilde{L}(\beta')-\tilde{L}(\beta'') \geq \langle\nabla\tilde{L}(\beta''), \beta'-\beta''\rangle+\alpha_3\|\beta'-\beta''\|_2^2 \quad (4.3)$$

for all $(\beta', \beta'') \in \{(\beta', \beta''):\sum_{i:\beta_{true,i}=0}\mathbb{I}(\beta_i'-\beta_i'' \neq 0) \leq s\}$ for some $\alpha_3 > 0$ and $s \geq k$. Evidently, this implies that (4.3) also holds for all $\|\beta'-\beta''\|_0 \leq s$.

To differentiate the two RSCs, we will refer to (4.2) as RSC_1, and to (4.3) as RSC_2. A closer observation reveals that both RSCs imply that the objective function of the nonconvex learning problem is strongly convex in some sparse subspace involving k number of dimensions.

Lemma 4.1: *Assume that $L(\beta)$ satisfies RSC_1 in (4.2). If $k \geq 1$, $\frac{64k\tau\log d}{n}+\mu \leq \alpha$, and $\mu\|\beta\|_2^2+\sum_{i=1}^{p} P_\lambda(|\beta_i|)$ is convex, then*

$$\frac{1}{n}\mathscr{L}(\beta')-\frac{1}{n}\mathscr{L}(\beta'') \geq \left\langle\frac{1}{n}\nabla\mathscr{L}(\beta''), \beta'-\beta''\right\rangle+\alpha_3\|\beta'-\beta''\|_2^2, \forall\|\beta''-\beta''\|_0 \leq s, \quad (4.4)$$

for some $\alpha_3 > 0$, where $s = 64k - 1$, $\nabla\mathscr{L}(\beta'') \in [n\nabla L(\tilde{\beta}) + n\lambda\partial|\beta|]_{\beta=\beta''}$, and $\partial|\beta|$ denotes the subdifferential of $|\beta|$.

The proof is given in the online supplement S3. From this lemma, we know that RSC_1, together with other assumptions made by Loh and Wainwright (2015), implies (4.4) for some $s \geq \|\beta_{true}\|_0 = k$ for all $k \geq 1$. Similarly for RSC_2, if the function L satisfy (4.3), in view of the the convexity of $\lambda|\beta|$, we have that $\frac{\mathscr{L}(\beta)}{n}=\tilde{L}(\beta)+\lambda|\beta|$ satisfies (4.4) for some $s \geq \|\beta_{true}\|_0$. In summary, (4.4) is a necessary condition to both RSC_1 and RSC_2.

Nonetheless, (4.4) can be restrictive in some scenarios. To illustrate this, we conduct a series of simulations as following: We simulated a sequence of samples $\{(x_t, y_t) : 1 \leq t \leq n\}$ randomly from the following sparse linear regression model: $y_t=x_t^\top\beta_{true}+\varepsilon_t$, for all $t = 1, \ldots, n$, in which d is set to 100, and $\beta_{true} = [1; 1; \mathbf{0}_{d-2}]$. Furthermore, $\varepsilon_t \sim N(0, 0.09)$ and $x_t \sim \mathcal{N}_d(0, \Sigma)$ for all $t = 1, \ldots, n$ with covariance matrix $\Sigma = (\sigma_{ij}) \in \mathbb{R}^{d\times d}$ defined as $\sigma_{ij} = \rho^{|i-j|}$. This numerical test considers only SCAD for an example. We set the parameters for the SCAD penalty as $a = 3.7$ and $\lambda = 0.2$.

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

We conduct a "random RSC test" to see if the randomly generated sample instances can satisfy the RSC condition. Notice that both versions of RSC dictate that the strong convexity be satisfied in a sparse subspace that has only k number of significant parameters. In this example, we have $k = 2$. Therefore, to numerically check if RSC is satisfied, we conduct the following procedures: (i) we randomly select two dimensions $i_1, i_2 : 1 \le i_1 < i_2 \le d$; (ii) we randomly sample two points $\beta^1, \beta^2 \in \{\beta \in \mathbb{R}^d : \beta_i = 0, \forall i \notin \{i_1, i_2\}\}$; and (iii) we check if a necessary condition for (4.4) holds. That is, we check if the following inequality holds, when $\beta^1 \neq \beta^2$:

$$\frac{\mathscr{L}(\beta^1)+\mathscr{L}(\beta^2)}{2} > \mathscr{L}\left(\frac{\beta^1+\beta^2}{2}\right). \quad (4.5)$$

We consider different sample sizes $n \in \{20, 25, 30, 35\}$ and the covariance matrix parameters $\rho \in \{0.1, 0.3, 0.5\}$ and constructed twelve sets of sample instances. Each set includes 100 random sample instances generated as mentioned above. For each sample instance, we conduct 10,000 repetitions of the "random RSC test". If (4.5) is satisfied for all these 10,000 repetitions, we say that the sample instance has passed the "random RSC test". Table 1 reports the test results.

We observe from Table 1 that in some cases the percentage for passing the random RSC test is noticeably low. However, with the increase of sample size, that percentage grows quickly. Moreover, we can also observe that when ρ is larger, it tends to be more difficult for RSC to hold. Figure 1 presents a typical instance that does not satisfy RSC when $n = 20$ and $\rho = 0.5$. This figure shows the 3-D contour plot of objective function when the decision variable is within the subspace $\{\beta : \beta_i = 0, \forall i \notin \{19, 20\}\}$. We can see that the contour plot apparently indicates nonconvexity of the function in the subspace, which violates (4.5).

We then compare MIPGO with both gradient methods in two sets of the sample instances from the table: (i) the one that seems to provide the most advantageous problem properties ($\rho = 0.1$, and $n = 35$) to the gradient methods; and (ii) the one with probably the most adversarial parameters ($\rho = 0.5$, and $n = 20$) to the gradient methods. Notice that the two gradient methods are implemented on MatLab following the descriptions by Wang, Liu and Zhang (2014) and Loh and Wainwright (2015), respectively, including their initialization procedures. MIPGO is also implemented on MatLab calling Gurobi (http://www.gurobi.com/). We use CVX, "a package for specifying and solving convex programs" (Grant and Boyd, 2013, 2008), as the interface between MatLab and Gurobi. Table 2 presents our comparison results in terms of computational, statistical and optimization measures. More specifically, we use the following criteria for our comparison:

- Absolute deviation (AD), defined as the distance between the computed solution and the true parameter vector. Such a distance is measured by ℓ_1 norm.
- False positive (FP), defined as the number of entries in the computed solution that are wrongly selected as nonzero dimensions.
- False negative (FN), defined as the number of entries in the computed solution that are wrongly selected as zero dimensions.

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

- objective gap ("Gap"), defined as the difference between the objective value of the computed solution and the objective value of the MIPGO solution. A positive value indicates a worse relative performance compared to MIPGO.
- Computational time ("Time"), which measures the total computational time to generate the solution.

AD, FP, and FN are commonly used statistical criteria, and "Gap" is a natural measure of optimization performance. In Table 2, we report the average values for all the above criteria out of 100 randomly generated instances aforementioned. From this table, we observe an outperformance of MIPGO over the other solution schemes on solution quality for both statistical and optimization criteria. However, MIPGO generates a higher computational overhead than the gradient methods.

5. Numerical Comparison on Optimization Performance with Local Linear Approximation

In this section, we numerically compare MIPGO with local linear approximation (LLA). We implement LLA on MatLab. In the implementation, we invoke the procedures of LLA iteratively until the algorithm fully converges. This shares the same spirit as the multistage procedure advocated by Huang and Zhang (2012). At each iteration, the LASSO subproblem is solved with Gurobi 6.0 using CVX (Grant and Boyd, 2013, 2008) as the interface. We report in the following a series of comparison results in terms of the optimization accuracy.

5.1. Numerical Tests on A Two-Dimensional Problem

In the following we conduct a numerical test on a two-dimensional LR-SCAD and a two-dimensional LR-MCP problem. We generate one instance for both of LR-SCAD and LR-MCP problems through the following procedures: We randomly generate $\beta_{true} \in \mathbb{R}^2$ with a uniformly distributed random vector on $[-1, 5]^2$ and then generate 2 observations $x_t \sim \mathcal{N}_2(0, \Sigma)$, $t \in \{1, 2\}$, with covariance matrix $\Sigma = (\sigma_{ij})$ and $\sigma_{ij} = 0.5^{|i-j|}$. Finally, we compute y_t as $y_t = x_t^\top \beta + \varepsilon_t$ with $\varepsilon_t \sim \mathcal{N}(0, 1)$ for all $t \in \{1, 2\}$. Both the LR-SCAD problem and the LR-MCP problem use the same set of samples $\{(x_t, y_t) : t = 1, 2\}$ in their statistical loss functions. The only difference between the two is the different choices of penalty functions. The parameters for the penalties are prescribed as $\lambda = 1$ and $a = 3.7$ for the SCAD and $\lambda = 0.5$ and $a = 2$ for the MCP. Despite their small dimensionality, these problems are nonconvex with multiple local solutions. Their nonconvexity can be visualized via the 2-D and 3-D contour plots provided in Figure 2(a)–(b) (LR-SCAD) and Figure 3(a)–(b) (LR-MCP).

We realize that the solution generated by LLA may depend on its starting point. Therefore, to make a fair numerical comparison, we consider two possible initialization procedures: (i) LLA with random initial solutions generated with a uniform distribution on the set $[0, 3.5]^2$ (denoted LLA_I), and (ii) LLA with initial solution set to be the least squares solution (denoted by LLA_{LSS}). (We will also consider LLA initialized with LASSO in later sections.)

To fully study the impact of initialization to the solution quality, we repeat each solution scheme 20 times. The best (Min.), average (Ave.), and worst (Max.) objective values as well

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

as the relative objective difference ($\text{gap}_{(\%)}$) obtained in the 20 runs are reported in Table 3. Here $\text{gap}_{(\%)}$ is defined

$$\frac{\{\text{Objective of computed solution}\}-\{\text{Objective of MIPGO solution}\}}{\{\text{Objective of computed solution}\}}\times 100\%.$$

From the table we have the following observations:

1. LLA_r's performance varies in different runs. In the best scenario, LLA attains the global optimum, while the average performance is not guaranteed.
2. LLA_{LSS} fails to attain the global optimal solution.
3. LLA with either initialization procedure yields a local optimal solution.
4. MIPGO performs robustly and attains the global solution at each repetition.

Figure 2(c) and Figure 3(c) present the search trajectories (dot dash lines) and convergent points (circles) of LLA_r for LR-SCAD and LR-MCP, respectively. In both figures, we observe a high dependency of LLA's performance on the initial solutions. Note that the least squares solutions for the two problems are denoted by the black squares. Figure 2(d) and Figure 3(d) present the convergent points of LLA_{LSS} for LR-SCAD and for LR-MCP, respectively. LLA_{LSS} utilizes the least squares solution (denoted as the black square in the figure) as its starting point. This least squares solution happens to be in the neighborhood of a local solution in solving both problems. Therefore, the convergent points out of the 20 repetitions of LLA_{LSS} all coincide with the least squares solution. Even though we have $n = d = 2$ in this special case, we can see that choosing the least squares solution as the initial solution may lead the LLA to a non-global stationary point. The solutions obtained by MIPGO is visualized in Figure 2(b) and 3(b) as triangles. MIPGO generates the same solution over the 20 repetitions even with random initial points.

5.2. Numerical Tests on Larger Problems

In the following, we conduct similar but larger-scale simulations to compare MIPGO and LLA in terms of optimization performance. For these simulations, we randomly generate problem instances as follows: We first randomly generate a matrix $T \in \mathbb{R}^{d\times d}$ with the entry on i-th row and j-th column uniformly distributed on $[0, 0.5^{|i-j|}]$ and set $\Sigma = T^\top T$ as the covariance matrix. Let the true parameter vector $\beta_{true} = [3\ 2\ 10\ 0\ 1\ 1\ 2\ 3\ 1.6\ 6\ \mathbf{0}_{1\times(d-10)}]$. We then randomly generate a sequence of observations $\{(x_t, y_t) : t = 1, \ldots, n\}$ following a linear model $y_t = x_t^\top \beta_{true} + \varepsilon_t$, where $x_t \sim \mathcal{N}_d(\mathbf{0}, \Sigma)$, and $\varepsilon_t \sim \mathcal{N}(0, 1.44)$ for all $t = 1, \ldots, n$. Finally, the penalty parameters are $\lambda = 1$ and $a = 3.7$ for SCAD and $\lambda = 0.5$ and $a = 2$ for MCP.

Following the aforementioned descriptions, we generate problem instances with different problem sizes d and sample sizes n (with 3 problem instances generated for each combination of d and n) and repeat each solution scheme 20 times. For these 20 runs, we randomly generate initial solutions for MIPGO with each entry following a uniform distribution on $[-10, 10]$. Similar to the 2-dimensional problems, we also involve in the comparison LLA with different initialization procedures: (i) LLA with randomly generated

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

initialization solution whose each entry follows a uniform distribution on [−10, 10] (denoted LLA_r). (ii) LLA with zero vector as the initial solution (denoted LLA_0). (iii) LLA with the initial solution prescribed as the solution to the LASSO problem (denoted LLA_1). More specifically, the LASSO problem used in the initialization of LLA_1 is formulated as

$$\min_{\beta\in\mathbb{R}^n}\frac{1}{2}\|y-X\beta\|_2^2+\omega\sum_{i=1}^{d}|\beta_i| \quad (5.1)$$

where $X := (x_1, \ldots, x_n)^\top$, $y := (y_1, \ldots, y_n)^\top$, and $\omega := \frac{1}{10}n\lambda\cdot(K-1)$ at K-th run with $1 \le K \le 20$. This is designed to examine how sensitive the performance the LLA_1 depends on the initial estimate. We would also like to remark that, when $K = 1$, the initial solution for LLA will be exactly the least squares solution. We would also like to remark that the LLA initialized with LASSO is the solution scheme proposed by Fan, Xue and Zou (2014).

The best (Min.), the average (Ave.), and the worst (Max.) objective values and the relative objective differences ($\text{gap}_{(\%)}$) of the 20 runs for each instance are reported in the upper and lower panels of Table 4 for LR-SCAD and LR-MCP, respectively. Notice that for each problem scale, we generate three test instances randomly, but Table 4 only reports one of the three instances for each problem size due to the limit of space. Tables S1 and S2 in Appendix S4 will complement the rest of the results. According to the numerical results, in all instances with different dimensions, MIPGO yields the lowest objective value, and in many cases, $\text{gap}_{(\%)}$ value is nontrivially large. This indicates the outperformance of our proposed MIPGO over all counterpart algorithms.

6. Numerical Comparison on Statistical Performance with Local Algorithms

We next examine MIPGO on the statistical performance in comparison with several existing local algorithms, including coordinate descent, LLA, and gradient methods. We simulate the random samples $\{(x_t, y_t), t = 1, \ldots, n\}$ from the following linear model:

$y_t = x_t^\top(\beta_{true,i}: 1 \le i \le d-1) + \beta_{true,d} + \varepsilon_t$, where we let $d = 1001$, $n = 100$, and $\beta_{true,d}$ is the intercept. β_{true} is constructed by first setting $\beta_{true,d} = 0$, then randomly choosing 5 elements among dimensions $\{1, \ldots, d-1\}$ to be 1.5, and setting the other $d-6$ elements as zeros. Furthermore, for all $t = 1, \ldots, n$, we let $\varepsilon_t \sim N(0, 1.44)$ and $x_t \sim \mathcal{N}_{d-1}(0, \Sigma)$ with $\Sigma = (\sigma_{ij})$ defined as $\sigma_{ij} = 0.5^{|i-j|}$. For both LR-SCAD and LR-MCP, we set the parameter $a = 2$, and tune λ the same way as presented by Fan, Xue and Zou (2014). We generate 100 instances using the above procedures, and solve each of these instances using MIPGO and other solutions schemes, including: (i) coordinate descent; (ii) gradient methods; (iii) SCAD-based and MCP-based LLA; and (iv) the LASSO method. The relative details of these techniques are summarized as follows:

LASSO: The LASSO penalized linear regression, coded in MatLab that invokes Gurobi 6.0 using CVX as the interface.

GM_1-SCAD/MCP: The SCAD/MCP penalized linear regression computed by the local solution method by Loh and Wainwright (2015) on Mat-Lab.

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

GM_2-SCAD/MCP: The SCAD/MCP penalized linear regression computed by the approximate path following algorithm by Wang, Liu and Zhang (2014) on MatLab.

SparseNet: The R-package *sparsenet* for SCAD/MCP penalized linear regression computed by coordinate descent (Mazumder et al., 2011).

Ncvreg-SCAD/-MCP: The R-package *ncvreg* for MCP penalized linear regression computed by coordinate descent (Breheny and Huang, 2011).

SCAD-LLA_1/MCP-LLA_1: The SCAD/MCP penalized linear regression computed by (fully convergent) LLA with the tuned LASSO estimator as its initial solution, following Fan, Xue and Zou (2014).

Notice that we no longer involve LLA_r and LLA_0 in this test, because a similar numerical experiment presented by Fan, Xue and Zou (2014) has shown that LLA_1 is more preferable than most other LLA variants in statistical performance.

Numerical results are presented in Table 5. According to the table, the proposed MIPGO approach estimates the (in)significant coefficients correctly in both SCAD and MCP penalties, and provides an improvement on the average AD over all the other alternative schemes.

To further measure the performance of different schemes, we use the oracle estimator as a benchmark. The oracle estimator is computed as following: Denote by $x_{t,i}$ as the i-th dimension of the t-th sample x_t, and by $\mathcal{S}$ the true support set, i.e., $\mathcal{S} := \{i : \beta_i^{true} \neq 0\}$. We conduct a linear regression using $\hat{X} := (x_{t,i} : t = 1, \ldots, n, i \in \mathcal{S})$ and $y := (y_t)$. As has been shown in Table 5, MIPGO yields a very close average AD and standard error to the oracle estimator. This observation is further confirmed in Figure 4. Specifically, Figures 4.(a) and 4.(b) illustrate relative the performance of LLA_1 and of MIPGO, respectively, in contrast to the oracle estimators. We see that MIPGO well approximates the oracle solution. Comparing MIPGO and LLA_1 from the figures, we can tell a noticeably improved recovery quality by MIPGO in contrast to LLA_1.

Nonetheless, we would like to remark that, although MIPGO yields a better solution quality over all the other local algorithms in every cases of the experiment as presented, the local algorithms are all noticeably faster than MIPGO. Therefore, we think that MIPGO is less advantageous in terms of computational time.

6.1. A Real Data Example

In this section, we conduct our last numerical test comparing MIPGO, LLA, and the gradient methods on a real data set collected in a marketing study (Wang, 2009; Lan et al., 2013), which has a total of $n = 463$ daily records. For each record, the response variable is the number of customers and the originally 6397 predictors are sales volumes of products. To facilitate computation, we employ the feature screening scheme in Li et al. (2012) to reduce the dimension to 1500. The numerical results are summarized in Table 6. In this table, GM_1 and GM_2 refer to the local solution methods proposed by Loh and Wainwright (2015) and by Wang, Liu and Zhang (2014), respectively. LLA_0 denote the LLA initialized as zero. LLA_1 denote the LLA initialized as the solution generated by LASSO. To tune the LASSO, we

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

implement LLA_1 choosing the coefficients ω in the LASSO problem (5.1) from the set $\{0.1\times nK\lambda : K = \{0, 1, \ldots, 20\}\}$ and we select the ω that enables LLA_1 to yield the best objective value. Here the value of λ is the same as the tuning parameter of SCAD or MCP. As reported in Table 6, $\lambda = 0.02$ for SCAD, and $\lambda = 0.03$ for MCP, respectively. Observations from Table 6 can be summarized as following: (i) for the case with the SCAD penalty, the proposed MIPGO yields a significantly better solution than all other alternative schemes in terms of both Akaike's information criterion (AIC), Bayesian information criterion (BIC), and the objective value. Furthermore, MIPGO also outputs a model with the smallest number of parameters. (ii) for the MCP case, both MIPGO and LLA_1 outperforms other schemes. Yet these two approaches have similar values for AIC and BIC. Nonetheless, MIPGO provides a better model as the number of nonzero parameters is smaller than the solution generated by LLA_1.

7. Conclusion

The lack of solution schemes that ascertain solution quality to nonconvex learning with folded concave penalty has been an open problem in sparse recovery. In this paper, we seek to address this issue in a direct manner by proposing a global optimization technique for a class of nonconvex learning problems without imposing very restrictive conditions.

In this paper, we provide a reformulation of the nonconvex learning problem into a general quadratic program. This reformulation then enables us to have the following findings:

- **a.** To formally state the complexity of finding the global optimal solution to the nonconvex learning with the SCAD and the MCP penalties.
- **b.** To derive a MIP-based global optimization approach, MIPGO, to solve the SCAD and MCP penalized nonconvex learning problems with theoretical guarantee. Numerical results indicate that the proposed MIPGO outperforms the gradient method by Loh and Wainwright (2015) and Wang, Liu and Zhang (2014) and LLA approach with different initialization schemes in solution quality and statistical performance.

To the best of our knowledge, the complexity bound of solving the nonconvex learning with the MCP and SCAD penalties globally has not been reported in literature and MIPGO is the first optimization scheme with provable guarantee on global optimality for solving a folded concave penalized learning problem.

We would like to alert the readers that the proposed MIPGO scheme, though being effective in globally solving the nonconvex learning with the MCP and SCAD penalty problem, yields a comparatively larger computational overhead than the local solution method in larger scale problems. (See comparison of computing times in Table 5.) In the practice of highly time-sensitive statistical learning with hugh problem sizes, LLA and other local solution schemes can work more efficiently. However, there are important application scenarios where a further refinement on the solution quality or even the exact global optimum is required. MIPGO is particularly effective in those applications, as it is the only method that is capable of providing the refinement with theoretical guarantee.

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

Finally, we would like to remark that the quadratic programming reformulation of penalized least squares with the MCP and SCAD penalty can be further exploited to develop convex approximation, complexity analyses, and solution schemes for finding a local solution. Those will be the future extensions of the presented work herein.

8. Proofs of Theorems 3.2 and 3.4

In this section, we give proofs of Theorems 3.2 and 3.4.

Proof of Theorem 3.2—Recall that **1** denotes an all-ones vector of a proper dimension. The program has a Lagrangian $\mathcal{F}_{SCAD}$ given as:

$$\begin{aligned}&\mathbb{F}_{SCAD}(\beta,g,h,\gamma_1,\gamma_2,\gamma_3,\gamma_4,\rho):\\&=\frac{1}{2}[\beta^\top Q\beta+n(a-1)g^\top g+2ng^\top h]\\&+q^\top\beta-na\lambda\mathbf{1}^\top g\\&+\gamma_1^\top(\beta\\&-h)-\gamma_2^\top(\beta+h)+\gamma_3^\top(-g)+\gamma_4^\top(g\\&\quad-\lambda\mathbf{1})\\&+\rho^\top(\mathscr{A}^\top\beta-\mathbf{b}),\end{aligned}$$

where $\gamma_1:=(\gamma_{1,i})\in\mathbb{R}^d_+$, $\gamma_2:=(\gamma_{2,i})\in\mathbb{R}^d_+$, $\gamma_3:=(\gamma_{3,i})\in\mathbb{R}^d_+$, $\gamma_4:=(\gamma_{4,i})\in\mathbb{R}^d_+$, and $\rho\in\mathbb{R}^m_+$ are Lagrangian multipliers. The KKT condition yields

$$\left\{\begin{array}{c}\frac{\partial\mathbb{F}_{SCAD}}{\partial\beta}:=Q\beta+q+\gamma_1-\gamma_2+\mathscr{A}\rho=0\\\frac{\partial\mathbb{F}_{SCAD}}{\partial h}:=ng-\gamma_1-\gamma_2=0\\\frac{\partial\mathbb{F}_{SCAD}}{\partial g}:=n(a-1)g+nh-na\lambda\mathbf{1}-\gamma_3+\gamma_4=0\end{array}\right. \tag{8.1}$$

$$\left\{\begin{array}{c}\gamma_{1,i}\geq 0;\gamma_{1,i}\cdot(\beta_i-h_i)=0\\\gamma_{2,i}\geq 0;\gamma_{2,i}\cdot(-\beta_i-h_i)=0\\\gamma_{3,i}\geq 0;\gamma_{3,i}\cdot g_i=0\\\gamma_{4,i}\geq 0;\gamma_{4,i}\cdot(g_i-\lambda)=0\\\rho\geq 0;\rho^\top(\mathscr{A}^\top\beta-\mathbf{b})=0,\end{array}\right\}\quad\forall i=1,\ldots,d \tag{8.2}$$

Since Λ is non-empty and $\mathcal{A}$ is full rank, it is easy to check that the linear independence constraint qualification is satisfied. Therefore, the global solution satisfies the KKT condition. This leads us to an equivalent representation of (2.4) in the form:

$$\min\ \frac{1}{2}[\beta^\top Q\beta+n(a-1)g^\top g+2ng^\top h]+q^\top\beta-na\lambda\mathbf{1}^\top g,\quad s.t. \tag{8.3}$$

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

$$\begin{cases} \beta \in \Lambda; \quad h \geq \beta; \quad h \geq -\beta; \quad 0 \leq g \leq \lambda \\ \text{Constraints } (8.1) - (8.2) \\ \beta,\, g,\, h,\, \gamma_1,\, \gamma_2,\, \gamma_3,\, \gamma_4 \in \mathbb{R}_+^d; \rho \in \mathbb{R}_+^m. \end{cases} \tag{8.4}$$

Then, it suffices to show that (8.3)–(8.4) is equivalent to (3.7)–(3.8).

Notice that the objective function (8.3) is immediately

$$\frac{1}{2}\beta^\top(Q\beta+q)+\frac{1}{2}q^\top\beta+\frac{1}{2}g^\top(n(a-1)g-n\alpha\lambda\mathbf{1}+2nh)-\frac{1}{2}na\lambda\mathbf{1}^\top g=:I_1$$

Due to equalities (8.1),

$$I_1=\frac{1}{2}\beta^\top(\gamma_2-\gamma_1)-\frac{1}{2}\beta^\top\mathscr{A}\rho+\frac{1}{2}q^\top\beta+\frac{1}{2}h^\top(\gamma_1+\gamma_2)+\frac{1}{2}g^\top(\gamma_3-\gamma_4)-\frac{1}{2}na\lambda\mathbf{1}^\top g \tag{8.5}$$

Invoking the complementarity conditions in (8.2), we may have

$$I_1=\frac{1}{2}q^\top\beta-\frac{1}{2}\mathbf{b}^\top\rho-\frac{1}{2}na\lambda\mathbf{1}^\top g-\frac{1}{2}\lambda\gamma_4^\top\mathbf{1}. \tag{8.6}$$

Therefore, Program (8.3)–(8.4) is equivalent to

$$\min\ I_1=\frac{1}{2}q^\top\beta-\frac{1}{2}\mathbf{b}^\top\rho-\frac{1}{2}na\lambda\mathbf{1}^\top g-\frac{1}{2}\lambda\gamma_4^\top\mathbf{1} \quad s.t. \quad (8.4), \tag{8.7}$$

which is immediately the desired result.

Proof of Theorem 3.4—The proof follows a closely similar argument as that for Theorem 3.4. The Lagrangian $\mathcal{F}_{MCP}$ of Program (2.6) yields:

$$\begin{aligned} &\mathbb{F}_{MCP}(\beta, g, h, \eta_1, \eta_2, \eta_3, \eta_4, \rho): \\ &\quad =\frac{1}{2}\beta^\top Q\beta \\ &\quad +q^\top\beta+n\frac{1}{2a}g^\top g \\ &\quad -n\left(\frac{1}{a}g-\lambda\mathbf{1}\right)^\top h \\ &\quad +\eta_1^\top(\beta \\ &\quad -h)+\eta_2^\top(-\beta-h)-\eta_3^\top g+\eta_4^\top(g \\ &\quad\quad -a\lambda\mathbf{1}) \\ &\quad +\rho^\top(\mathscr{A}^\top\beta-\mathbf{b}) \end{aligned} \tag{8.8}$$

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

where **1** denotes an all-ones vector of a proper dimension, and where $\eta_4 \in \mathbb{R}_+^d$ and $\rho \in \mathbb{R}_+^m$ are Lagrangian multipliers. The KKT condition yields

$$\begin{cases} \frac{\partial \mathbb{F}_{MCP}}{\partial \beta} := q + Q\beta + \eta_1 - \eta_2 + \mathscr{A}\rho = 0 \\ \frac{\partial \mathbb{F}_{MCP}}{\partial g} := \frac{n}{a} g - \frac{n}{a} h - \eta_3 + \eta_4 = 0 \\ \frac{\partial \mathbb{F}_{MCP}}{\partial h} := -n\left(\frac{1}{a} g - \lambda \mathbf{1}\right) - \eta_1 - \eta_2 = 0 \end{cases} \tag{8.9}$$

$$\begin{cases} \eta_1^\top (\beta - h) = 0;\ \eta_2^\top (-\beta - h) = 0 \\ \eta_3^\top g = 0;\ \eta_4^\top (g - a\lambda \mathbf{1}) = 0;\ \rho^\top (\mathscr{A}^\top \beta - \mathbf{b}) = 0 \\ \eta_1 \geq 0;\ \eta_2 \geq 0;\ \eta_3 \geq 0;\ \eta_4 \geq 0;\ \rho \geq 0. \end{cases} \tag{8.10}$$

Since Λ is non-empty and $\mathscr{A}$ is full rank, it is easily verifiable that the linear independence constraint qualification is satisfied. This means the KKT system holds at the global solution. Therefore, Imposing additional constraints (8.9)–(8.10) in program (2.6) will not result in inequivalence. Notice that the object function in (2.6) equals

$$I_2 := \frac{1}{2} q^\top \beta + \left(\frac{1}{2} q^\top + \frac{1}{2} \beta^\top Q\right)\beta + \frac{n}{2a} g^\top (g - h) - \frac{1}{2} n \left(\frac{1}{a} g^\top - \lambda \mathbf{1}^\top\right) h + \frac{1}{2} \lambda n \mathbf{1}^\top h \tag{8.11}$$

Per (8.9), we obtain

$$I_2 = \frac{1}{2} q^\top \beta - \frac{1}{2} (\eta_1 - \eta_2 + \mathscr{A}\rho)^\top \beta + \frac{1}{2} g^\top (\eta_3 - \eta_4) + \frac{1}{2} (\eta_1 + \eta_2)^\top h + \frac{1}{2} \lambda n \mathbf{1}^\top h \tag{8.12}$$

Further noticing (8.10) we obtain

$$\begin{aligned} I_2 &= \tfrac{1}{2} q^\top \beta - \tfrac{1}{2} \mathbf{b}^\top \rho + \tfrac{1}{2} g^\top (\eta_3 - \eta_4) + \tfrac{1}{2} \lambda n \mathbf{1}^\top h \\ &= \tfrac{1}{2} q^\top \beta - \tfrac{1}{2} \mathbf{b}^\top \rho - \tfrac{1}{2} g^\top \eta_4 + \tfrac{1}{2} \lambda n \mathbf{1}^\top h \\ &= \tfrac{1}{2} q^\top \beta - \tfrac{1}{2} \mathbf{b}^\top \rho - \tfrac{1}{2} a\lambda \mathbf{1}^\top \eta_4 + \tfrac{1}{2} \lambda n \mathbf{1}^\top h \end{aligned}$$

which immediately leads to the desired result.

Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

References

Bertsimas, D.; Chang, A.; Rudin, C. Integer optimization methods for supervised ranking. 2011. Available at http://hdl.handle.net/1721.1/67362

Breheny P, Huang J. Coordinate descent algorithms for nonconvex penalized regression, with applications to biological feature selection. Annals of Applied Statistics. 2011; 5:232–253. [PubMed: 22081779]

Fan J, Li R. Variable selection via nonconcave penalized likelihood and its oracle properties. Journal of American Statistical Association. 2001; 96:1348–1360.

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

Fan J, Lv J. Non-concave penalty likelihood with NP-dimensionality. IEEE Transections on Information Theory. 2011; 57:5467–5484.

Fan J, Peng H. Nonconcave penalized likelihood with a diverging number of parameters. The Annals of Statistics. 2004; 32:928–961.

Fan, J.; Xue, L.; Zou, H. Strong oracle optimality of folded concave penalized estimation. The Annals of Statistics. 2012. Available at http://arxiv.org/pdf/1210.5992v1.pdf

Fan J, Xue L, Zou H. Strong oracle optimality of folded concave penalized estimation. The Annals of Statistics. 2014; 42:819–849.

Huang J, Zhang CH. Estimation and selection via absolute penalized convex minimization and its multistage adaptive applications. Journal of Machine Learning Research. 2012; 13:1839–1864. [PubMed: 24348100]

Hunter D, Li R. Variable selection using MM algorithms. The Annals of Statistics. 2005; 33:1617–1642.

Kim Y, Choi H, Oh HS. Smoothly clipped absolute deviation on high dimensions. Journal of American Statistical Association. 2008; 103:1665–1673.

Lan, W.; Zhong, P-S.; Li, R.; Wang, H.; Tsai, C-L. Working paper. 2013. Testing a single regression coefficient in high dimensional linear models.

Lawler EL, Wood DE. Branch-and-bound methods: a survey. Operations Research. 1966; 14(4):699–719.

Li R, Zhong W, Zhu L. Feature screening via distance correlation. Journal of the American Statistical Association. 2012; 107(499):1129–1139. [PubMed: 25249709]

Liu H, Yao T, Li R. Supplement to “Global solutions to folded concave penalized nonconvex learning”. 2014

Loh, P-L.; Wainwright, MJ. Regularized M-estimators with nonconvexity: statistical and algorithmic theory for local optima. Journal of Machine Learning Research. 2015. To appear. Available at http://arxiv.org/pdf/1305.2436.pdf

Grant, M.; Boyd, S. CVX: Matlab software for disciplined convex programming, version 2.0 beta. 2013. http://cvxr.com/cvx

Grant, M.; Boyd, S. Recent advances in learning and control. Springer; London: 2008. Graph implementations for nonsmooth convex programs; p. 95-110.

Martí, R.; Reinelt, G. The Linear Ordering Problem. Springer; Berlin Heidelberg: 2011. Branch-and-bound; p. 85-94.

Mazumder R, Friedman J, Hastie T. SparseNet Coordinate descent with non-convex penalties. Journal of American Statistical Association. 2011; 106:1125–1138.

Meinshausen N, Bühlmann P. High-dimensional graphs and variable selection with the lasso. The Annals of Statistics. 2006; 34:1436–1462.

Nesterov, Y. CORE Discussion Papers 2007076. Universit Catholique de Louvain, Center for Operations Research and Dconometrics (CORE); 2007. Gradient methods for minimizing composite objective function.

Pardalos PM. Global optimization algorithms for linearly constrained indefinite quadratic problems. Computers & Mathematics with Applications. 1991; 21(6–7):87–97.

Vandenbussche D, Nemhauser GL. A polyhedral study of nonconvex quadratic programs with box constraints. Mathematical Programming. 2005; 102:531–557.

Vavasis, SA. International Series of Monographs on Computer Science. Oxford Science Publications; 1991. Nonlinear Optimization: Complexity Issues.

Vavasis SA. Approximating algorithms for indefinite quadratic programming. Mathematical Programming. 1992; 57:279–311.

Wang H. Forward regression for ultra-high dimensional variable screening. Journal of the American Statistical Association. 2009; 104:1512–1524.

Wang L, Kim Y, Li R. Calibrating nonconvex penalized regression in ultrahigh dimension. The Annals of Statistics. 2013; 41:2505–2536.

Wang Z, Liu H, Zhang T. Optimal computational and statistical rates of convergence for sparse nonconvex learning problems. The Annals of Statistics. 2014; 42:2164–2201.

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

Zhang C. Nearly unbiased variable selection under minimax concave penalty. The Annals of Statistics. 2010; 28:894–942.

Zhang C, Zhang T. A general theory of concave regularization for high dimensional sparse estimation problems. Statistical Science. 2012 To appear.

Zou H. The adaptive lasso and its oracle properties. Journal of American Statitical Association. 2006; 101:1418–1429.

Zou H, Li R. One-step sparse estimation in non-concave penalized likelihood models. The Annals of Statistics. 2008; 36:1509–1533.

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

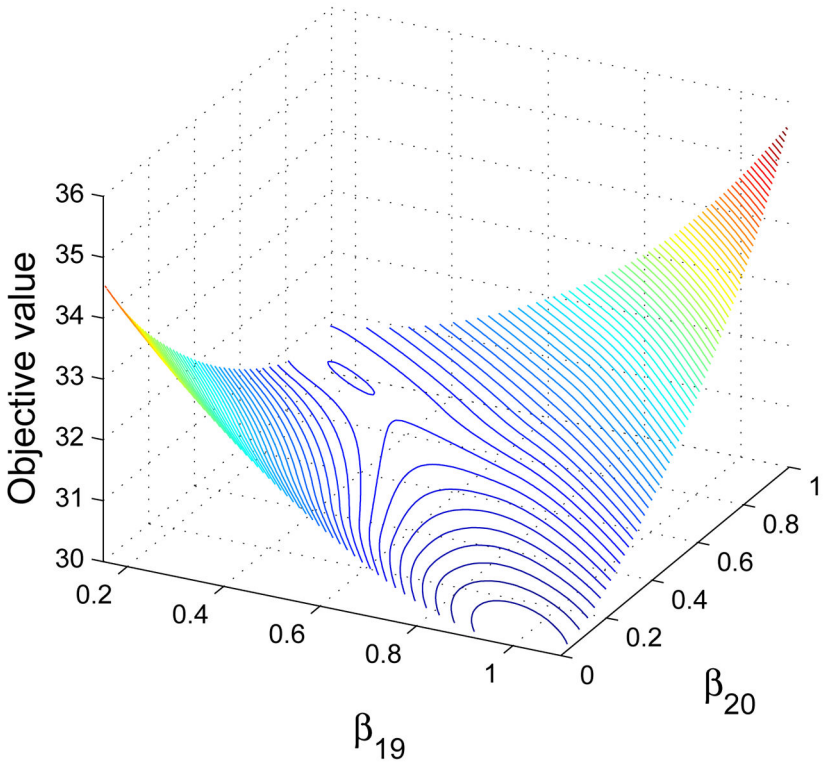

Fig. 1.
A sample instance that fails in the random RSC test.

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

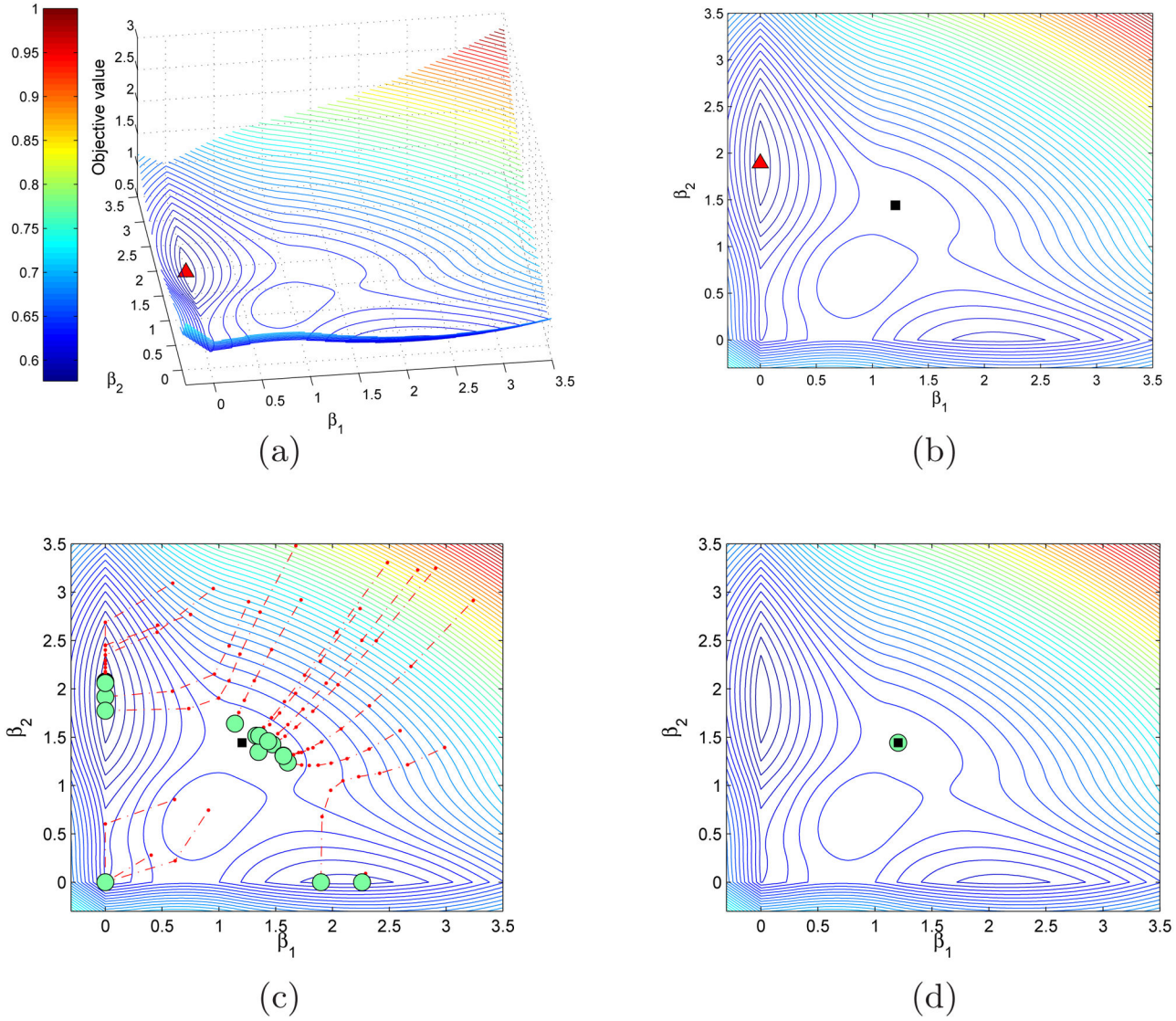

Fig. 2.

(a) 3-D contour plots of the 2-dimension LR-SCAD problem and the solution generated by MIPGO in 20 runs with random initial solutions. The triangle is the MIPGO solution in both subplots. (b) 2-D representation of subplot (a). (c) Trajectories of 20 runs of LLA with random initial solutions. (d) Trajectories of 20 runs of LLA with the least squares solution as the initial solutions.

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

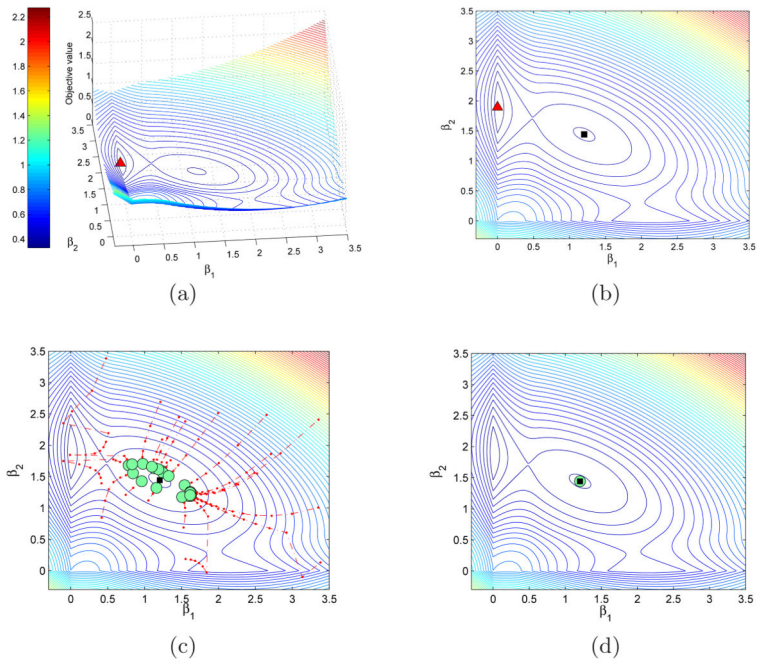

Fig. 3.

(a). 3-D contour plots of the 2-dimension LR-MCP problem and the solution generated by MIPGO in 20 runs with random initial solutions. The triangle is the MIPGO solution in both subplots. (b). 2-D representation of subplot (a). (c). Trajectories of 20 runs of LLA with random initial solutions. (d). Trajectories of 20 runs of LLA with the least squares solution as the initial solutions.

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

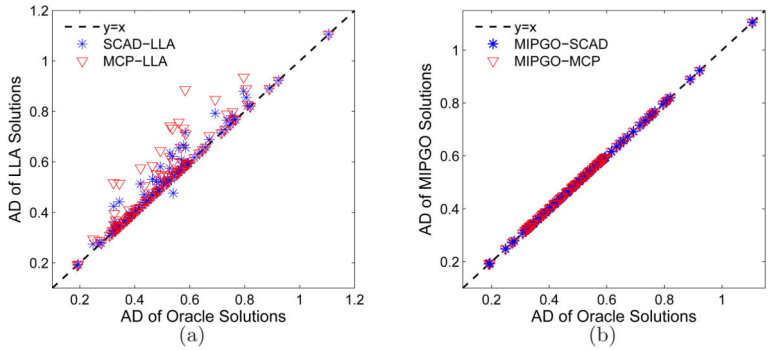

Fig. 4.

Comparison between generated solutions and the oracle solutions in AD when (a) solutions are generated by LLA_1, and (b) solutions are generated by MIPGO. The horizontal axis is the AD value of the oracle solution for each simulation, while the vertical axis is the AD of generated solutions for the same simulation. The closer a point is to the linear function "$y = x$", the smaller is the difference between the AD of a generated solution and the AD of the corresponding oracle solution.

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

Table 1

Percentage for successfully passing the random RSC test out of 100 randomly generated instances.

ρ	$n = 35$	$n = 30$	$n = 25$	$n = 20$
0.1	93%	81%	53%	4%
0.3	94%	76%	39%	9%
0.5	55%	50%	21%	1%

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

Table 2

Comparison between MIPGO and the gradient methods. The numbers in parenthesis are the standard errors. GM_1 and GM_2 stand for the gradient methods proposed by Loh and Wainwright (2015) and Wang, Liu and Zhang (2014), respectively.

Method	AD	FP	FN	Gap	Time
	$\rho = 0.5$, $n = 20$				
MIPGO	0.188 (0.016)	0.230 (0.042)	0 (0)	0 (0)	29.046 (5.216)
GM_1	2.000 (0.000)	0 (0)	2 (0)	25.828 (0.989)	0.002 (0.001)
GM_2	0.847 (0.055)	5.970 (0.436)	0 (0)	1.542 (0.119)	0.504 (0.042)
	$\rho = 0.1$, $n = 35$				
MIPGO	0.085 (0.005)	0.020 (0.141)	0 (0)	0 (0)	27.029 (4.673)
GM_1	2.000 (0.000)	0 (0)	2 (0)	31.288 (1.011)	0.002 (0.000)
GM_2	0.936 (0.044)	6.000 (0.348)	0 (0)	4.179 (0.170)	0.524 (0.020)

Ann Stat. Author manuscript; available in PMC 2017 April 01.

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

Table 3

Test result on a toy problem. "$\text{gap}_{(\%)}$" stands for the relative difference in contrast to MIPGO.

Penalty	Measure	LLA_r	$\text{gap}_{(\%)}$	LLA_{LSS}	$\text{gap}_{(\%)}$	MIPGO
SCAD	Min	0.539	0.00	0.900	40.12	0.539
	Ave	0.911	40.85	0.900	40.12	0.539
	Max	2.150	74.93	0.900	40.12	0.539
MCP	Min	0.304	2.63	0.360	17.78	0.296
	Ave	0.435	31.95	0.360	17.78	0.296
	Max	1.293	77.11	0.360	17.78	0.296

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

Table 4

Numerical comparison of LLA and the proposed MIPGO on LR-SCAD and LR-MCP problems with different problem scales. “TS” stands for “Typical Sample”.

		MIPGO	LLA_r	$gap_{(\%)}$	LLA_0	$gap_{(\%)}$	LLA_1	$gap_{(\%)}$
LR-SCAD								
TS 3	Min.	89.87	89.87	0.00	104.96	14.37	104.96	14.37
d = 10	Ave.	89.87	109.19	17.69	104.96	14.37	104.96	14.37
n = 10	Max.	89.87	162.93	44.84	104.96	14.37	104.96	14.37
TS 6	Min.	86.04	88.219	2.46	115.17	25.30	108.37	20.60
d = 20	Ave.	86.04	105.13	18.15	115.17	25.30	108.37	20.60
n = 10	Max.	86.04	143.51	40.05	115.17	25.30	108.37	20.60
TS 9	Min.	120.35	120.35	0.00	150.15	19.85	120.35	0.00
d = 40	Ave.	120.35	167.21	28.02	150.15	19.85	120.35	0.00
n = 15	Max.	120.35	203.18	40.76	150.15	19.85	120.35	0.00
TS 12	Min.	519.14	519.14	0.00	560.28	7.34	538.47	3.59
d = 200	Ave.	519.14	733.06	29.18	560.28	7.34	538.47	3.59
n = 60	Max.	519.14	959.00	45.87	560.28	7.34	538.47	3.59
TS 15	Min.	841.72	841.90	0.02	1003.69	16.14	873.44	3.63
d = 500	Ave.	841.72	981.73	14.26	1003.69	16.14	873.44	3.63
n = 80	Max.	841.72	1173.06	28.25	1003.69	16.14	873.44	3.63
TS 18	Min.	1045.22	1105.70	5.47	1119.84	6.66	1119.84	6.66
d = 1000	Ave.	1045.22	1135.84	7.98	1119.84	6.66	1119.84	6.66
n = 100	Max.	1045.22	1309.70	20.19	1119.84	6.66	1119.84	6.66
LR-MCP								
TS 3	Min.	13.65	15.77	13.43	21.51	36.54	25.00	45.39
d = 10	Ave.	13.65	20.60	39.59	21.51	36.54	25.00	45.39
n = 10	Max.	13.65	32.83	58.41	21.51	36.54	25.00	45.39

Ann Stat. Author manuscript; available in PMC 2017 April 01.

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

		MIPGO	LLA_r	gap(%)	LLA_0	gap(%)	LLA_1	gap(%)
TS 6	Min.	14.71	17.60	16.41	14.71	0.00	20.06	26.67
d = 20	Ave.	14.71	17.60	22.54	14.71	0.00	20.06	26.67
n = 10	Max.	14.71	17.60	65.38	14.71	0.00	20.06	26.67
TS 9	Min.	23.64	27.17	13.02	26.57	11.05	49.08	51.84
d = 40	Ave.	23.64	27.17	35.40	26.57	11.05	49.08	51.84
n = 15	Max.	23.64	27.17	57.98	26.57	11.05	49.08	51.84
TS 12	Min.	93.55	105.62	11.42	112.13	16.57	120.63	22.45
d = 200	Ave.	93.55	165.25	43.39	112.13	16.57	120.63	22.45
n = 60	Max.	93.55	596.52	84.32	112.13	16.57	120.63	22.45
TS 15	Min.	163.98	175.44	6.53	221.84	26.08	179.53	8.66
d = 500	Ave.	163.98	211.62	22.51	221.84	26.08	179.53	8.66
n = 80	Max.	163.98	237.56	30.97	221.84	26.08	179.53	8.66
TS 18	Min.	249.89	267.83	6.70	267.83	6.70	272.39	8.27
d = 1000	Ave.	249.89	322.24	22.25	267.83	6.70	272.39	8.27
n = 100	Max.	249.89	530.60	52.60	267.83	6.70	272.39	8.27

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

Table 5

Comparison of statistical performance. “Time” stands for the computational time in second. The numbers in parenthesis are the standard errors.

Method	$n = 100, d = 1000$			
	AD	FP	FN	Time
LASSO	2.558 (0.047)	5.700 (0.255)	0 (0)	2.332 (0.108)
GM_1-SCAD	0.526 (0.017)	0.600 (0.084)	0 (0)	4.167 (0.254)
GM_1-MCP	0.543 (0.018)	0.540 (0.073)	0 (0)	4.42 (0.874)
GM_2-SCAD	3.816 (0.104)	18.360 (0.655)	0 (0)	3.968 (0.049)
GM_2-MCP	0.548 (0.019)	0.610 (0.083)	0 (0)	3.916 (0.143)
SparseNet	1.012 (0.086)	5.850 (1.187)	0 (0)	2.154 (0.017)
Ncvreg-SCAD	1.068 (0.061)	9.220 (0.979)	0 (0)	0.733 (0.007)
Ncvreg-MCP	0.830 (0.045)	3.200 (0.375)	0 (0)	0.877 (0.009)
SCAD-LLA_1	0.526 (0.017)	0.600 (0.084)	0 (0)	31.801 (1.533)
MCP-LLA_1	0.543 (0.018)	0.540 (0.073)	0 (0)	28.695 (1.473)
MIPGO-SCAD	0.509 (0.017)	0 (0)	0 (0)	472.673 (97.982)
MIPGO-MCP	0.509 (0.017)	0 (0)	0 (0)	361.460 (70.683)
Oracle	0.509 (0.017)			

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

Table 6

Results of the Real Data Example. “NZ”, $\#_{0.05}$, and $\#_{0.10}$ stand for the numbers of parameters that are nonzero, that has a p-value greater or equal to 0.05, and that has a p-value greater or equal to 0.1. “R^2” denotes the R-squared value. “AIC”, “BIC” and “Obj.” stand for Akaike’s information criterion, Bayesian information criterion and objective function value.

Method	NZ	$\#_{0.05}$	$\#_{0.10}$	R^2	AIC	BIC	Obj.
			SCAD: $\lambda = 0.02$; $a = 3.7$				
GM_1	1500	—	—	0.997	357.101	6563.691	212.279
GM_2	401	401	401	0.698	246.886	1906.114	103.626
LLA_0	185	119	115	0.864	−554.093	211.387	76.031
LLA_1	181	83	80	0.912	−763.718	14.789	71.673
MIPGO	129	35	34	0.898	−796.581	−262.814	68.474
			MCP: $\lambda = 0.03$; $a = 2$				
GM_1	818	—	—	0.332	1448.966	5129.436	169.091
GM_2	134	110	104	0.735	−296.624	−346.474	93.870
LLA_0	96	5	6	0.856	704.654	−307.432	72.645
LLA_1	113	2	2	0.902	−849.842	−382.279	69.292
MIPGO	109	3	3	0.899	−841.280	−390.267	68.591