Neurocomputing 119 (2013) 413–424

Contents lists available at SciVerse ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

A comparison of typical ℓ_p minimization algorithms

Qin Lyu [a], Zhouchen Lin [a], Yiyuan She [b], Chao Zhang [a,*]

[a] Key Laboratory of Machine Perception (MOE), School of EECS, Peking University, PR China
[b] Department of Statistics, Florida State University, USA

ARTICLE INFO

Article history:
Received 18 September 2012
Received in revised form
21 January 2013
Accepted 5 March 2013
Communicated by D. Cai
Available online 3 May 2013

Keywords:
Compressed sensing
Sparse representation
ℓ_1 minimization
ℓ_p minimization

ABSTRACT

Recently, compressed sensing has been widely applied to various areas such as signal processing, machine learning, and pattern recognition. To find the sparse representation of a vector w.r.t. a dictionary, an ℓ_1 minimization problem, which is convex, is usually solved in order to overcome the computational difficulty. However, to guarantee that the ℓ_1 minimizer is close to the sparsest solution, strong incoherence conditions should be imposed. In comparison, nonconvex minimization problems such as those with the ℓ_p $(0<p<1)$ penalties require much weaker incoherence conditions and smaller signal to noise ratio to guarantee a successful recovery. Hence the ℓ_p $(0<p<1)$ regularization serves as a better alternative to the popular ℓ_1 one. In this paper, we review some typical algorithms, *Iteratively Reweighted ℓ_1 minimization* (IRL1), *Iteratively Reweighted Least Squares* (IRLS) (and its general form *General Iteratively Reweighted Least Squares* (GIRLS)), and *Iteratively Thresholding Method* (ITM), for ℓ_p minimization and do comprehensive comparison among them, in which IRLS is identified as having the best performance and being the fastest as well.

1. Introduction

Compressed sensing [1,2] has drawn much attention in recent years. It has found wide applications in various areas such as signal processing, machine learning, and pattern recognition. At the core of the compressed sensing theory, one has to solve for the sparsest representation vector of a given vector with respect to a given dictionary:

$$\min_{\mathbf{x}\in\mathbb{R}^n}\|\mathbf{x}\|_0, \quad \text{subject to } \mathbf{y}=\mathbf{A}\mathbf{x}, \tag{Eℓ0}$$

where $\mathbf{A}\in\mathbb{R}^{m\times n}$ $(m\ll n)$ is an over-complete dictionary, $\mathbf{y}$ is a given vector, and $\|\mathbf{x}\|_0$ is the number of non-zeros in $\mathbf{x}$.

Unfortunately, problem Eℓ_0 is NP-hard [3] for general $\mathbf{A}$ and $\mathbf{y}$. To overcome such a computational difficulty, various methods have been proposed. A major class of methods is to solve

$$\min_{\mathbf{x}\in\mathbb{R}^n}\|\mathbf{x}\|_1, \quad \text{subject to } \mathbf{y}=\mathbf{A}\mathbf{x}, \tag{Eℓ1}$$

instead and Donoho [4] proved that under some conditions problem (Eℓ_1) is equivalent to (Eℓ_0) with an overwhelming probability. Many algorithms have been proposed to solve (Eℓ_1). Heuristic greedy algorithms include Orthogonal Matching Pursuit (OMP) [5] and Least Angle Regression (LARS) [6]. As (Eℓ_1) is convex, many efficient algorithms that guarantee globally optimal solutions have also been proposed, such as Gradient Projection (GP) [7], Homotopy [8], Iterative Shrinkage-Thresholding (IST) [9], Accelerated Proximal Gradient (APG) [10], and Alternating Direction Method (ADM) [11] and its linearized version (LADM) [12]. Interested readers may refer to [13] for a comprehensive comparison among these algorithms. Other excellent reviews on ℓ_1 minimization algorithms include [14,15].

However, some conditions [4], e.g., the sparsest solution is indeed very sparse and the matrix $\mathbf{A}$ satisfies low coherence conditions, are necessary to guarantee the equivalence between (Eℓ_1) and (Eℓ_0). But in practice, these conditions may not be satisfied. So solving (Eℓ_1) may fail to provide a desired solution. In this case, one has to turn to other nonconvex variants of (Eℓ_0), which requires weaker conditions to guarantee a successful recovery, to solve for the sparsest solution. One natural variant is via ℓ_p minimization:

$$\min_{\mathbf{x}\in\mathbb{R}^n}\|\mathbf{x}\|_p^p, \quad \text{subject to } \mathbf{y}=\mathbf{A}\mathbf{x}, \tag{Eℓp}$$

where $0<p<1$. It is intuitive that when p is close to 0, the solution to (Eℓ_p) will be close to that of (Eℓ_0), hence producing a sparser solution than (Eℓ_1). This has been supported by theoretical analysis [16,17]. So it is desirable to develop efficient algorithms for (Eℓ_p).

When there is noise, as in ℓ_1 minimization the following variants of (Eℓ_p) are also often considered:

$$\min_{\mathbf{x}\in\mathbb{R}^n}\|\mathbf{x}\|_p^p, \quad \text{subject to } \|\mathbf{y}-\mathbf{A}\mathbf{x}\|_2^2\le\varepsilon, \tag{Nℓp}$$

* Corresponding author.
E-mail addresses: lvqin@pku.edu.cn (Q. Lyu), zlin@pku.edu.cn (Z. Lin), yshe@stat.fsu.edu (Y. She), chzhang@cis.pku.edu.cn (C. Zhang).

0925-2312/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.neucom.2013.03.017

and

$$\min_{\mathbf{x}\in\mathbf{R}^n} \frac{1}{2}\|\mathbf{y}-\mathbf{A}\mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|_p^p. \quad \text{(L}\ell\text{p)}$$

In practice (Lℓ_p) is more popular than (Nℓ_p). So like [13] in this paper we mainly focus on (Eℓ_p) and (Lℓ_p).

A major drawback of ℓ_p minimization is that it is nonconvex. So it is challenging to design efficient algorithms for solving (Eℓ_p) and (Lℓ_p). Nonetheless, there is still much effort devoted to solving (Eℓ_p) and (Lℓ_p) efficiently, although not as abundant as those for (Eℓ_1) and (Lℓ_1). There have been various algorithms targeting on ℓ_p minimization, e.g., [16–28], to name just a few. However, we have found that many of them are actually equivalent or have only slight difference between each other. Actually, the existing algorithms can be categorized into three kinds: *Iteratively Reweighted ℓ_1 minimization* (IRL1), *Iteratively Reweighted Least Squares* (IRLS) (and its general form *General Iteratively Reweighted Least Squares* (GIRLS)), and *Iteratively Thresholding Method* (ITM). This paper aims at briefly reviewing these three representative ℓ_p minimization algorithms and systematically comparing their performance.

Although for nonconvex problems no globally optimal solutions can always be guaranteed and the convergence of the algorithms is much more difficult to analyze, these algorithms empirically work well for ℓ_p minimization.

The remainder of this paper is organized as follows. Section 2 will introduce three typical algorithms for ℓ_p minimization problems. Section 3 will compare ℓ_p minimization and ℓ_1 minimization and the typical algorithms for ℓ_p minimization. Finally, Section 4 concludes this paper.

2. Algorithms

In this section we will introduce three typical algorithms for ℓ_p minimization problems: *Iteratively Reweighted ℓ_1 minimization* (IRL1), *Iteratively Reweighted Least Squares* (IRLS) (and its general form *General Iteratively Reweighted Least Squares* (GIRLS)), and *Iteratively Thresholding Method* (ITM). IRL1 and IRLS can solve both the constrained problem Eℓ_p and the unconstrained problem Lℓ_p, while ITM is designed for the unconstrained problem only.

We use bold font, like $\mathbf{x}$, to denote a vector and denote x_i for its component, i.e., $\mathbf{x}=(x_1,x_2,\ldots,x_n)\in\mathbb{R}^n$. Especially, we denote the true solution as $\mathbf{x}_0$. Moreover, we denote the value of a variable at the lth iteration as $(\cdot)^{(l)}$, e.g., $\mathbf{x}^{(l)}$ represents the solution obtained at the lth iteration. Finally, $\mathbf{x}^*$ denotes the converged solution.

2.1. Iteratively reweighted ℓ_1 minimization

2.1.1. For equality constrained problem

Since the IRL1 algorithm for ℓ_p minimization is derived from reweighted ℓ_1 minimization algorithm [29], we introduce the latter first.

Consider the following weighted ℓ_1 minimization problem:

$$\min_{\mathbf{x}\in\mathbb{R}^n} \sum_{i=1}^n w_i|x_i|, \quad \text{subject to } \mathbf{y}=\mathbf{A}\mathbf{x}. \tag{1}$$

To improve the sparsity of the solution, Candes et al. [29] suggested that the weights be chosen as inversely proportional to the magnitudes of the components of the true solution, i.e.,

$$w_i = \begin{cases} \dfrac{1}{|x_{0,i}|}, & x_{0,i}\neq 0, \\ +\infty, & x_{0,i}=0. \end{cases} \tag{2}$$

Since $\mathbf{x}_0$ is unknown when we solve (1), and to avoid division by zeros, Candes et al. [29] suggested that the weights be chosen according to the current iterate:

$$w_i^{(l+1)} = \frac{1}{|x_i^{(l)}|+\varepsilon}, \tag{3}$$

where $0<\varepsilon\ll 1$ is a small parameter to prevent division by zeros. It is shown in [29] that by iteratively solving the weighted ℓ_1 minimization problem, with the weights chosen as (3), sparser solutions can be obtained than by directly solving the ℓ_1 minimization problem (Eℓ_1).

When extending the reweighted ℓ_1 algorithm for ℓ_p minimization, several papers, like [18–20], all suggested using the following weights:

$$w_i^{(l+1)} = \frac{1}{(|x_i^{(l)}|+\varepsilon_l)^{1-p}}, \tag{4}$$

where $\{\varepsilon_l\}$ is a sequence of positive real numbers that approach zero to avoid the problem of division by zeros. In practice one can also set it to be a small positive constant. Then the iteration goes as follows:

$$\mathbf{x}^{(l+1)} = \operatorname*{argmin}_{\mathbf{x}\in\mathbb{R}^n} \sum_{i=1}^n \frac{|x_i|}{(|x_i^{(l)}|+\varepsilon^{(l)})^{1-p}}, \quad \text{subject to } \mathbf{y}=\mathbf{A}\mathbf{x}. \tag{5}$$

Namely, IRL1 solves a series of ℓ_1 minimization problems to approximate the minimizer of ℓ_p minimization problem. The pseudo code of IRL1 for the equality constrained problem is presented in Algorithm 1.

Algorithm 1. Iteratively reweighted ℓ_1 minimization algorithm (for equality constrained problem)

Input: $\mathbf{A}\in\mathbb{R}^{m\times n}, \mathbf{y}\in\mathbb{R}^m, p\in(0,1)$.

1: Set a sequence of positive numbers $\{\varepsilon_l\}$ such that $\lim_{l\to\infty}\varepsilon_l=0$. Initialize $\mathbf{x}^{(0)}$ such that $\mathbf{y}=\mathbf{A}\mathbf{x}^{(0)}$.
2: **while** not converged ($l=0,1,2,\ldots$) **do**
3: Solve the following ℓ_1 minimization problem:
$\mathbf{x}^{(l+1)}\leftarrow\operatorname*{argmin}_{\mathbf{x}\in\mathbb{R}^n} \sum_{i=1}^n \frac{|x_i|}{(|x_i^{(l)}|+\varepsilon_l)^{1-p}}$, subject to $\mathbf{y}=\mathbf{A}\mathbf{x}$.4:
end while
5: $\mathbf{x}^*\leftarrow\mathbf{x}^{(l)}$.

Output: $\mathbf{x}^*$.

Some scholars have analyzed the convergence of IRL1 [30,19], but the results are all weak. However, numerical experiments showed that the iterates converge with an overwhelming probability, and actually converge to the sparsest solution when the measurements are sufficient, i.e., m/n is large enough.

2.1.2. For unconstrained problem

For unconstrained ℓ_p minimization problem (Lℓ_p), Gasso et al. [20] proposed updating $\mathbf{x}$ by the following way:

$$\mathbf{x}^{(l+1)} = \operatorname*{argmin}_{\mathbf{x}\in\mathbb{R}^n} \frac{1}{2}\|\mathbf{y}-\mathbf{A}\mathbf{x}\|_2^2 + \sum_{i=1}^n \lambda_i^{(l)}|x_i|, \tag{6}$$

where $\lambda_i^{(l)}=\lambda p/(|x_i^{(l)}|+\varepsilon_l)^{1-p}$ and ε_l is a small positive number.

Zou and Li also proposed and analyzed the same form in [28], where they called it *Local Linear Approximation* (LLA). The intuitive idea of both [20,28] is almost the same, which is to approximate the penalty function with its first-order Taylor expansion at the current iterate. Here we will follow [20], where the approximation was deduced in the framework of Difference of Convex functions (DC) programming [31].

For a nonconvex minimization problem:

$$\min_{\mathbf{x}\in\mathbb{R}^n} J(\mathbf{x}), \tag{7}$$

where $J(\cdot)$ is a nonconvex function, the main idea of DC programming is to decompose $J(\mathbf{x})$ as:

$$J(\mathbf{x}) = J_1(\mathbf{x}) - J_2(\mathbf{x}), \tag{8}$$

where $J_1(\cdot)$ and $J_2(\cdot)$ are lower semi-continuous proper convex functions on $\mathbb{R}^n$. Then $\mathbf{x}$ is updated as follows[1]:

$$\mathbf{x}^{(l+1)} = \operatorname*{argmin}_{\mathbf{x}} J_1(\mathbf{x}) - \langle \mathbf{y}^{(l)}, \mathbf{x} - \mathbf{x}^{(l)} \rangle, \tag{9}$$

where $\mathbf{y}^{(l)} \in \partial J_2(\mathbf{x}^{(l)})$ and $J_2(\mathbf{x})$ is approximated by its "first-order Taylor expansion" $J_2(\mathbf{x}^{(l)}) + \langle \mathbf{y}^{(l)}, \mathbf{x} - \mathbf{x}^{(l)} \rangle$ at $\mathbf{x}^{(l)}$, in which $\partial J_2(\cdot)$ is the subgradient of J_2. Then by choosing $J_1(\mathbf{x}) = \frac{1}{2}\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|_1$ and $J_2(\mathbf{x}) = \lambda(\|\mathbf{x}\|_1 - \|\mathbf{x}\|_p^p)$, we can obtain the following updating scheme:

$$\mathbf{x}^{(l+1)} = \operatorname*{argmin}_{\mathbf{x}\in\mathbb{R}^n} \frac{1}{2}\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \sum_{i=1}^{n} \frac{\lambda p}{|x_i^{(l)}|^{1-p}} |x_i|. \tag{10}$$

By introducing a small positive number ε_l to avoid division by zero, we can obtain (6).

Problem (6) is an adaptive LASSO problem [33], and can be solved by many methods for convex minimization, e.g., Approximate Proximal Gradient (APG) [10]. It can also be solved by the iteratively thresholding method which we will introduce in Section 2.3, where the thresholding function should be

$$\Phi_{l_1}(t;\lambda) = \begin{cases} t - \lambda, & \text{if } t > \lambda, \\ t + \lambda, & \text{if } t < -\lambda, \\ 0, & \text{if } |t| \le \lambda, \end{cases}$$

and the iteration goes as follows:

$$x_i^{(l+1)} = \Phi_{l_1}(t_i; \lambda_i^{(l)}\|\mathbf{A}\|_2^{-2}), \quad i = 1, 2, \ldots, n, \tag{11}$$

where t_i is the ith component of $\mathbf{t} = (\mathbf{I} - \|\mathbf{A}\|_2^{-2}\mathbf{A}^T\mathbf{A})\mathbf{x}^{(l)} + \|\mathbf{A}\|_2^{-2}\mathbf{A}^T\mathbf{y}$.

The pseudo code of IRL1 for unconstrained problem is summarized in Algorithm 2.

Algorithm 2. Iteratively reweighted ℓ_1 minimization algorithm (for unconstrained problem)

Input: $\mathbf{A} \in \mathbb{R}^{m\times n}, \mathbf{y} \in \mathbb{R}^m, p \in (0,1), \lambda > 0$.

1: Set a sequence of positive numbers $\{\varepsilon_l\}$ such that $\lim_{l\to\infty}\varepsilon_l = 0$. Initialize $\mathbf{x}^{(0)}$.
2: **while** not converged ($l = 0, 1, 2, \ldots$) **do**
3: Solve the following ℓ_1 minimization problem:
$\mathbf{x}^{(l+1)} \leftarrow \operatorname*{argmin}_{\mathbf{x}\in\mathbb{R}^n} \frac{1}{2}\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \sum_{i=1}^{n} \frac{\lambda p}{(|x_i^{(l)}| + \varepsilon_l)^{1-p}} |x_i|$.4:
end while
5: $\mathbf{x}^* \leftarrow \mathbf{x}^{(l)}$.

Output: $\mathbf{x}^*$.

For fixed ε_l, Chen and Zhou [30] proved that under some conditions the iterates of IRL1 converge to the global minimizer of a truncated ℓ_p minimization problem and the convergence rate is approximately linear. For variable ε_l, no theoretical analysis is available.

2.2. Iteratively reweighted least squares

2.2.1. For equality constraint problem

Iteratively reweighted least squares (IRLS) was proposed by Rao and Kreutz-Delgado in [34] for ℓ_p minimization. As stated in that paper, IRLS is equivalent to the FOCUSS algorithm originally proposed in [35,36]. Further discussions on IRLS can be found in [27,26,37].

[1] Here we switch to the deduction by the concave-convex procedure [32], which is equivalent to the deduction in [20] yet much more intuitive.

The formulation of IRLS is very similar to that of IRL1. However, the solution method is completely different. IRLS is essentially composed of a series of weighted ℓ_2 optimization problems as follows:

$$\min_{\mathbf{x}\in\mathbb{R}^n} \sum_{i=1}^{n} w_i x_i^2, \quad \text{subject to } \mathbf{y} = \mathbf{A}\mathbf{x}, \tag{12}$$

where the weights are set iteratively by

$$w_i^{(l)} = ((x_i^{(l)})^2 + \varepsilon_l)^{p/2-1}. \tag{13}$$

Here $\{\varepsilon_l\} \to 0$ is a sequence of positive numbers to avoid division by zeros. The above formulation was derived through the *minimizing a concave function via a convex function replacement* (MCCR) algorithm which was proposed by Mourad and Reilly [25]. The MCCR algorithm is to replace the objective function with a convex function, in particular, a quadratic function.

Let $\mathbf{Q}_l = \operatorname{diag}(\{1/w_i^{(l)}\})$, then the solution to (12) can be explicitly given as follows:

$$\mathbf{x}^{(l+1)} = \mathbf{Q}_l\mathbf{A}^T(\mathbf{A}\mathbf{Q}_l\mathbf{A}^T)^{-1}\mathbf{y}. \tag{14}$$

The original weighted least squares problem (12) can be rewritten as follows:

$$\min_{\mathbf{x}\in\mathbb{R}^n} \|\mathbf{Q}_l^{-1/2}\mathbf{x}\|_2^2, \quad \text{subject to } \mathbf{y} = \mathbf{A}\mathbf{x} \tag{15}$$

Mourad and Reilly suggested a more general form in [25], called Generalized IRLS (GIRLS), which is to update $\mathbf{x}$ by solving:

$$\min_{\mathbf{x}\in\mathbb{R}^n} \|\mathbf{Q}_l^{-1/2}(\mathbf{x} - \theta\mathbf{x}^{(l)})\|_2^2, \quad \text{subject to } \mathbf{y} = \mathbf{A}\mathbf{x}, \tag{16}$$

instead, where $\theta \in (0,1)$. Accordingly, the update scheme is as follows:

$$\mathbf{x}^{(l+1)} = \theta\mathbf{x}^{(l)} + (1-\theta)\mathbf{Q}_l\mathbf{A}^T(\mathbf{A}\mathbf{Q}_l\mathbf{A}^T)^{-1}\mathbf{y}. \tag{17}$$

Mourad and Reilly showed that the direction from $\mathbf{x}^{(l)}$ to $\mathbf{Q}_l\mathbf{A}^T(\mathbf{A}\mathbf{Q}_l\mathbf{A}^T)^{-1}\mathbf{y}$ provides a descending direction of the objective function $\|\mathbf{x}\|_p^p$ [25]. So $\theta \le 1$ is necessary for $\|\mathbf{x}^{(l+1)}\|_p^p \le \|\mathbf{x}^{(l)}\|_p^p$. Based on these facts, we can determine the "optimal" θ at each iteration by solving the following 1-D optimization problem:

$$\theta_l = \operatorname*{argmin}_{\theta_{min} < \theta < 1} \|\theta\mathbf{x}^{(l)} + (1-\theta)\mathbf{Q}_l\mathbf{A}^T(\mathbf{A}\mathbf{Q}_l\mathbf{A}^T)^{-1}\mathbf{y}\|_p^p, \tag{18}$$

where $\theta_{min} > 0$ is a lower bound of θ.

A brief summary of the IRLS algorithm for equality constrained problem is presented in Algorithm 3, and the GIRLS algorithm is summarized in Algorithm 4.

Algorithm 3. Iteratively reweighted least squares algorithm (for equality constrained problem)

Input: $\mathbf{A} \in \mathbb{R}^{m\times n}, \mathbf{y} \in \mathbb{R}^m, p \in (0,1)$.

1: Set a sequence of positive numbers $\{\varepsilon_l\}$ such that $\lim_{l\to\infty}\varepsilon_l = 0$. Initialize $\mathbf{x}^{(0)}$ such that $\mathbf{y} = \mathbf{A}\mathbf{x}^{(0)}$.
2: **while** not converged ($l = 0, 1, 2, \ldots$) **do**
3: Solve the following ℓ_1 minimization problem: Solve the following weighted least square problem:
$\mathbf{x}^{(l+1)} \leftarrow \arg\min_{x\in\mathbb{R}^n} \sum_{i=1}^{n} w_i^{(l)} x_i^2$, subject to $\mathbf{y} = \mathbf{A}\mathbf{x}$, (19)
where the weights are
$w_i^{(l)} = ((x_i^{(l)})^2 + \varepsilon_l)^{p/2-1}$.
(Or directly compute $\mathbf{x}^{(l+1)} \leftarrow \mathbf{Q}_l\mathbf{A}^T(\mathbf{A}\mathbf{Q}_l\mathbf{A}^T)^{-1}\mathbf{y}$, where $\mathbf{Q}_l = \operatorname{diag}(\{1/w_i^{(l)}\})$.)
4: **end while**
5: $\mathbf{x}^* \leftarrow \mathbf{x}^{(l)}$.

Output: $\mathbf{x}^*$.

Algorithm 4. Generalized Iteratively Reweighted Least Squares Algorithm (for equality constrained problem)

Input: $\mathbf{A} \in \mathbb{R}^{m\times n}, \mathbf{y} \in \mathbb{R}^m, p \in (0,1)$.

1: Set a sequence of positive numbers $\{\varepsilon_l\}$ such that $\lim_{l\to\infty}\varepsilon_l = 0$. Initialize $\mathbf{x}^{(0)}$ such that $\mathbf{y} = \mathbf{A}\mathbf{x}^{(0)}$.
2: **while** not converged ($l = 0, 1, 2, \ldots$) **do**
3: $\hat{\mathbf{x}}^{(l+1)} \leftarrow \mathbf{Q}_l\mathbf{A}^T(\mathbf{A}\mathbf{Q}_l\mathbf{A}^T)^{-1}\mathbf{y}$, where $\mathbf{Q}_l = \mathrm{diag}(\{1/w_i^{(l)}\})$.
4: Solve the following optimization problem:
$\theta_l \leftarrow \arg\min_{\theta} \|\theta\mathbf{x}^{(l)} + (1-\theta)\hat{\mathbf{x}}^{(l+1)}\|_p^p$, subject to $\theta_{min} < \theta < 1.5$:
$\mathbf{x}^{(l+1)} \leftarrow \theta_l\mathbf{x}^{(l)} + (1-\theta_l)\hat{\mathbf{x}}^{(l+1)}$.
6: **end while**
7: $\mathbf{x}^* \leftarrow \mathbf{x}^{(l)}$.
Output: $\mathbf{x}^*$.

Chartrand and Yin [26] analyzed the convergence behavior of Algorithm 3 and proved that if every $2\|\mathbf{x}_0\|_0$ columns of $\mathbf{A}$ is linearly independent then $\{\mathbf{x}^{(l)}\}$ converges to a vector whose sparsity is also $\|\mathbf{x}_0\|_0$. This result shows that IRLS is theoretically better than IRL1, whose convergence is uncertain. This will be verified by our experiments.

2.2.2. For unconstrained problem

The unconstrained ℓ_p optimization problem (Lℓ_p) can be approximated as

$$\min_{\mathbf{x}\in\mathbb{R}^n} \lambda\sum_{i=1}^{n}(x_i^2+\varepsilon)^{p/2} + \frac{1}{2}\|\mathbf{y}-\mathbf{A}\mathbf{x}\|_2^2 \doteq L_p(\mathbf{x},\varepsilon). \tag{20}$$

Consider the function $L_p(\mathbf{x},\varepsilon)$, its critical point $\mathbf{x}$ should satisfy the following equation:

$$\left[\frac{\lambda p x_i}{(\varepsilon+(x_i^2))^{1-p/2}}\right]_{1\le i\le n} + \mathbf{A}^T(\mathbf{A}\mathbf{x}-\mathbf{y}) = 0. \tag{21}$$

So Lai and Wang [21] suggested the following iteration scheme:

$$\left[\frac{\lambda p x_i^{(l+1)}}{(\varepsilon+(x_i^{(l)})^2)^{1-p/2}}\right]_{1\le i\le n} + \mathbf{A}^T(\mathbf{A}\mathbf{x}^{(l+1)}-\mathbf{y}) = 0, \tag{22}$$

or equivalently:

$$\left(\mathbf{A}^T\mathbf{A} + \mathrm{diag}\left(\left\{\frac{p\lambda}{(\varepsilon+(x_i^{(l)})^2)^{1-p/2}}\right\}\right)\right)\mathbf{x}^{(l+1)} = \mathbf{A}^T\mathbf{y}. \tag{23}$$

Note that the $\mathbf{x}^{(l+1)}$ obtained by (23) is also the minimizer of the following problem:

$$\min_{\mathbf{x}\in\mathbb{R}^n}\frac{1}{2}\|\mathbf{y}-\mathbf{A}\mathbf{x}\|_2^2 + \sum_{i=1}^{n}\frac{p\lambda}{(\varepsilon+(x_i^{(l)})^2)^{1-p/2}}x_i^2. \tag{24}$$

It is easy to see that the ℓ_p penalty function in (20) is approximated by a quadratic term in (24), which shares the same idea as *Local Quadratic Approximation* (LQA) [38]. Fan and Li applied LQA to several penalties like ℓ_0, ℓ_1, and SCAD in [38]. Later Hunter and Li studied the convergence property of the LQA algorithm and found that LQA is one of the minorize–maximize (MM) algorithms [39]. They also suggested a perturbed version like (24) for general penalty functions.

Lai and Wang [21] proved that $\{\mathbf{x}^{(l)}\}$ generated by IRLS converges to a critical point of (20). So one can get an approximate local minimizer of the unconstrained ℓ_p minimization problem by IRLS. To improve numerical accuracy, one may allow ε to change along iterations. The corresponding IRLS algorithm is summarized in Algorithm 5 and recently Lai et al. generalized the analysis on IRLS with a fixed ε to that with adaptive ε_l's [40].

Algorithm 5. Iteratively reweighted least squares algorithm (for unconstrained problem)

Input: $\mathbf{A}\in\mathbb{R}^{m\times n}, \mathbf{y}\in\mathbb{R}^m, p\in(0,1), \lambda > 0$.
1: Set a sequence of positive numbers $\{\varepsilon_l\}$ such that $\lim_{l\to\infty}\varepsilon_l = 0$. Initialize $\mathbf{x}^{(0)}$.
2: **while** not converged ($l = 0, 1, 2, \ldots$) **do**
3: Solve the following linear system:
$$\left(\mathbf{A}^T\mathbf{A} + \mathrm{diag}\left(\left\{\frac{p\lambda}{(\varepsilon_l+(x_i^{(l)})^2)^{1-p/2}}\right\}\right)\right)\mathbf{x} = \mathbf{A}^T\mathbf{y}.$$
and set $\mathbf{x}^{(l+1)} \leftarrow \mathbf{x}$.
4: **end while**
5: $\mathbf{x}^* \leftarrow \mathbf{x}^{(l)}$.
Output: $\mathbf{x}^*$.

Although the iterates of IRLS may converge to a sparse solution, they themselves may not have zero entries at all, because IRLS is composed of a series of ridge regression problems. Usually the output of IRLS contains a few large entries and a lot of entries with very small magnitudes. Thresholding may be adopted to enforce sparsity of the solution by IRLS. However, care must be taken in order to choose the threshold appropriately.

2.3. Iteratively thresholding method

Iteratively thresholding method (ITM) [41] is for unconstrained problem only. To introduce it, we start from a more general penalized regression problem as follows:

$$\min_{\mathbf{x}\in\mathbb{R}^n}\frac{1}{2}\|\mathbf{y}-\mathbf{A}\mathbf{x}\|_2^2 + P(\mathbf{x};\lambda) \doteq f(\mathbf{x}), \tag{25}$$

where $P(\mathbf{x};\lambda)$ is a penalty function. By introducing an auxiliary variable $\mathbf{z}$, we define

$$g(\mathbf{x},\mathbf{z}) = \tfrac{1}{2}\|\mathbf{y}-\mathbf{A}\mathbf{x}\|_2^2 + P(\mathbf{x};\lambda) + \tfrac{1}{2}(\mathbf{x}-\mathbf{z})^T(\mathbf{I}-\mathbf{A}^T\mathbf{A})(\mathbf{x}-\mathbf{z}). \tag{26}$$

Suppose $\mathbf{A}$ has been scaled properly such that $\|\mathbf{A}\|_2 < 1$, where $\|\mathbf{A}\|_2$ denotes the spectral norm (the largest singular value) of $\mathbf{A}$, then $\mathbf{I}-\mathbf{A}^T\mathbf{A}$ is positive definite and hence it is easy to see that minimizing $g(\mathbf{x},\mathbf{z})$ over $(\mathbf{x},\mathbf{z})$ is equivalent to minimizing $f(\mathbf{x})$ over $\mathbf{x}$.

We may minimize $g(\mathbf{x},\mathbf{z})$ by alternating minimization. Given $\mathbf{z}^{(l)}$, the update scheme for $\mathbf{x}^{(l+1)}$ can be found as equivalent to

$$\mathbf{x}^{(l+1)} = \underset{\mathbf{x}\in\mathbb{R}^n}{\mathrm{argmin}}\tfrac{1}{2}\|\mathbf{x}-[(\mathbf{I}-\mathbf{A}^T\mathbf{A})\mathbf{z}^{(l)} + \mathbf{A}^T\mathbf{y}]\|_2^2 + P(\mathbf{x};\lambda). \tag{27}$$

Given $\mathbf{x}^{(l+1)}$, minimizing over $g(\mathbf{x}^{(l+1)},\mathbf{z})$ simply gives $\mathbf{z}^{(l+1)} = \mathbf{x}^{(l+1)}$. Suppose the solution to (27) is given by

$$\mathbf{x}^{(l+1)} = \Phi((\mathbf{I}-\mathbf{A}^T\mathbf{A})\mathbf{z}^{(l)} + \mathbf{A}^T\mathbf{y};\lambda), \tag{28}$$

where Φ is called the thresholding function [41]. Then the iterations for solving (25) can be written as

$$\mathbf{x}^{(l+1)} = \Phi((\mathbf{I}-\mathbf{A}^T\mathbf{A})\mathbf{x}^{(l)} + \mathbf{A}^T\mathbf{y};\lambda). \tag{29}$$

She [41] showed that different penalty functions may result in the same thresholding function. It is shown in [23] that for the $\lambda\|\mathbf{x}\|_p^p$ penalty function, the corresponding thresholding function is

$$\Phi_{l_p}(t;\lambda) = \begin{cases} 0, & \text{if } |t|\le\tau(\lambda),\\ \mathrm{sgn}(t)\max\{\theta : g(\theta) = |t|\}, & \text{if } |t| > \tau(\lambda),\end{cases} \tag{30}$$

where $g(\theta;\lambda) = \theta + \lambda p\theta^{p-1}$, $\tau(\lambda) = \lambda^{1/(2-p)}(2-p)[p/(1-p)^{1-p}]^{1/(2-p)}$. Obviously, g attains its minimum $\tau(\lambda)$ at $\theta_0 = \lambda^{1/(2-p)}[p(1-p)]^{1/(2-p)}$. What is more, $g(\theta)$ is strictly increasing on $[\theta_0,+\infty)$ and $g(\theta)\to+\infty$ as $\theta\to+\infty$. Therefore, given any $t > \tau(\lambda)$, the equation $g(\theta) = t$ has one and only one root in $[\theta_0,+\infty)$, which can be found via any numerical method.

She [23] proved that if $\|\mathbf{A}\|_2 < 1$ then the algorithm converges to a stationary point of the objective function in (25). Therefore, we should use the following updating scheme to obtain a convergent

sequence:

$$\mathbf{x}^{(l+1)} = \Phi_{l_p}((\mathbf{I}-\|\mathbf{A}\|_2^{-2}\mathbf{A}^T\mathbf{A})\mathbf{x}^{(l)} + \|\mathbf{A}\|_2^{-2}\mathbf{A}^T\mathbf{y}; \lambda\|\mathbf{A}\|_2^{-2}). \quad (31)$$

A brief summary of ITM is in Algorithm 6.

ITM can also be done in an elementary-wise way, i.e., the entries of $\mathbf{x}$ are updated successively by fixing other entries, thus at each update we are solving a 1-dimensional ℓ_p minimization problem. This is actually the well-known coordinate descent algorithm [42], which has been proposed for LASSO for some time. The popular R package `glmnet` is based on this approach [43].

Algorithm 6. Iteratively thresholding method (for unconstrained problem)

Input: $\mathbf{A}\in\mathbb{R}^{m\times n}, \mathbf{y}\in\mathbb{R}^m, p\in(0,1), \lambda>0$.

1: Initialize $\mathbf{x}^{(0)}$.
2: **while** not converged ($l=0,1,2,\ldots$) **do**
3: $\mathbf{x}^{(l+1)}\leftarrow\Phi_{l_p}((\mathbf{I}-\|\mathbf{A}\|_2^{-2}\mathbf{A}^T\mathbf{A})\mathbf{x}^{(l)} + \|\mathbf{A}\|_2^{-2}\mathbf{A}^T\mathbf{y}; \lambda\|\mathbf{A}\|_2^{-2})$.
4: **end while**
5: $\mathbf{x}^*\leftarrow\mathbf{x}^{(l)}$.

Output: $\mathbf{x}^*$.

She proved that such an algorithm with no stepsize search always has a global convergence property [23], and it also works for group ℓ_p penalty in generalized linear models (including classification). Note that neither IRL1 nor IRLS has the guarantee of convergence without extra conditions. Another issue we want to mention here is that much faster convergence can be achieved by using a relaxation form [23], but we restrict ourselves to the basic form in this paper.

2.4. A brief summary

We have reviewed three types of algorithms for solving the ℓ_p minimization problems. Namely, iteratively reweighted ℓ_1 minimization (IRL1), iteratively reweighted least squares (IRLS) and its general form (GIRLS), and iteratively thresholding method (ITM). Here we give a brief summary on the relations and differences among them.

IRL1 and IRLS have close connections. They approximate the ℓ_p norm with the ℓ_1 and ℓ_2 norm, which result in solving a series of reweighted ℓ_1 and ℓ_2 problems, respectively. ITM, on the other hand, tackles the problem from a quite different way by using threshold function techniques.

One of their differences is the sparsity of solutions. Theoretically, reweighted ℓ_2 problems do not generate sparse solutions. Thus the solution may not have zero entries at all, even if the iterates of IRLS converge to a sparse vector. On the other hand, IRL1 and ITM guarantee the sparsity of each iterate, which makes their zero entries more convincing than the zeros obtained by the extra thresholding needed by IRLS.

Another difference is in their convergence properties. The conditions for the convergence of IRL1 and IRLS are rather strict, while ITM has a global convergence property. This makes the programming of ITM easier than those of IRL1 and IRLS.

3. Numerical experiments

In this section we present extensive experiments to compare the performance of ℓ_p minimization algorithms. The codes are all in MATLAB and run on a Dell workstation with dual quad-core 2.26 GHz Xeon processors and 24 GB of memory.

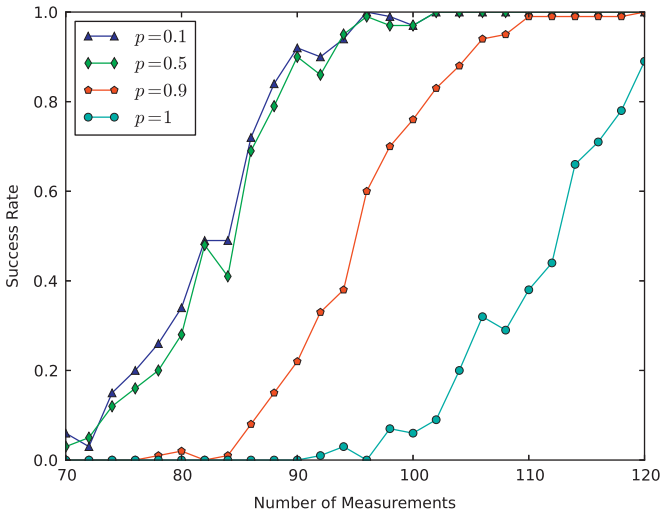


Fig. 1. Comparison of sparse recovery ability for ℓ_p minimization with different p's ($n=256, S=40, p=0.1, 0.5, 0.9, 1$). The horizontal and vertical axes are the number of measurements and the success rate, respectively.

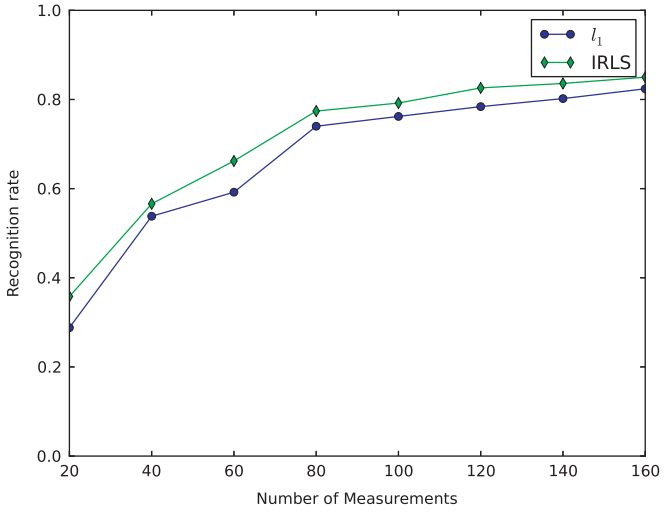


Fig. 2. The recognition rates under different projection dimensions. $n=190$, $p=0.5$, and m varies from 20 to 160.

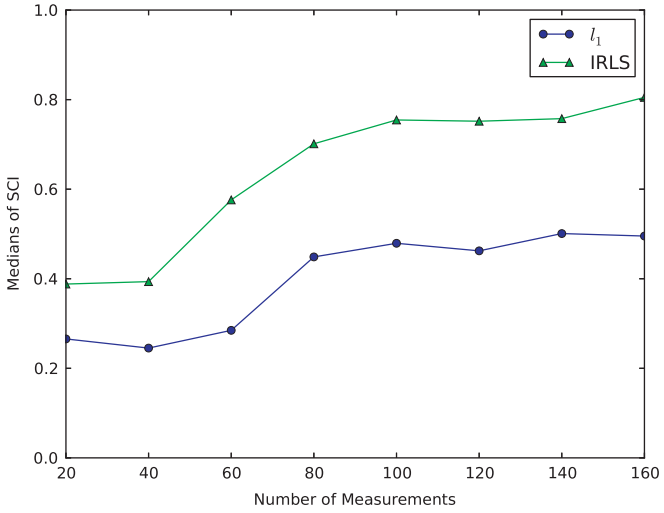


Fig. 3. Medians of SCI values under different projection dimensions. $n=190$, $p=0.5$, and m varies from 20 to 160.

3.1. Comparison of sparse recovery properties of ℓ_p and ℓ_1 minimizations

Intuitively and as analyzed in [16,17], ℓ_p minimization usually obtains sparser solutions than ℓ_1 minimization does. Moreover, the smaller p is, the sparser solution is. We will testify to this by experiments.

We fix vector length $n=256$ and sparsity $S=40$. Let the number m of measurements vary from 70 to 120. For each m, we generate a Gaussian random matrix $\mathbf{A}$ and normalize its columns to unit ℓ_2 length. Then we randomly generate a ground truth vector $\mathbf{x}_0$ and get the measurement vector $\mathbf{y}=\mathbf{A}\mathbf{x}_0$. Finally, we solve ℓ_p minimization problem Eℓ_p for $p=0.1, 0.5, 0.9$ by the IRLS algorithm. For ℓ_1 minimization problem Eℓ_1, we solve it by the primal-dual interior point method, which has a MATLAB implementation in the ℓ_1-magic package [44]. If the solution $\mathbf{x}^*$ satisfy $\|\mathbf{x}^*-\mathbf{x}_0\|_2/\|\mathbf{x}_0\|_2 < 10^{-3}$, we regard it as successful recovery. For each m we do experiments 100 times and calculate the successful recovery rate. The results are shown in Fig. 1.

As we can see from Fig. 1, ℓ_p minimization does have a much higher success rate in recovering the sparsest solution than ℓ_1 does, no matter $p=0.1, 0.5$ or 0.9. For fix p, the successful recovery rate grows as the number of measurements grows. And as p

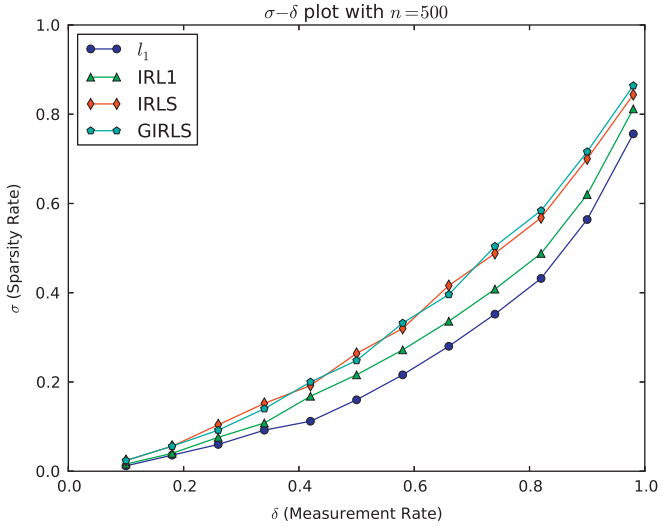


Fig. 4. The 90% success-rate curves of ℓ_1 and ℓ_p algorithms (p=0.5).

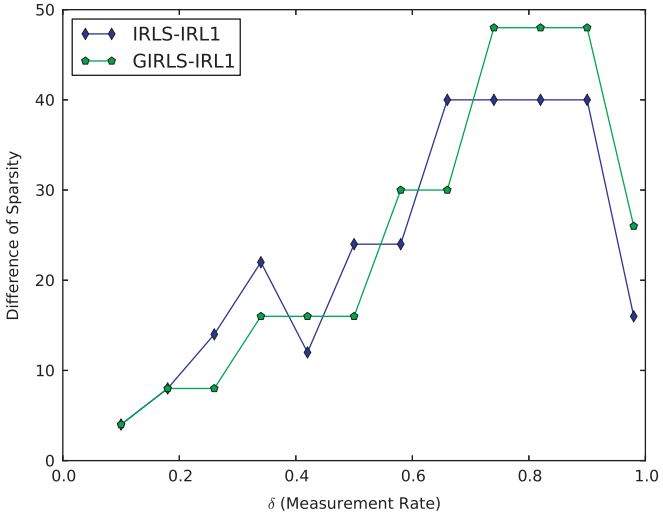


Fig. 5. Difference of sparsity under 90% success rate.

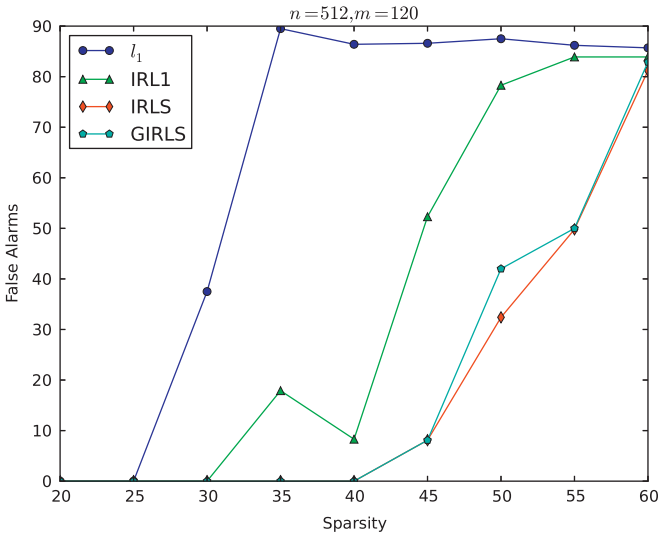


Fig. 6. False alarms of ℓ_1 and ℓ_p algorithms (p=0.5).

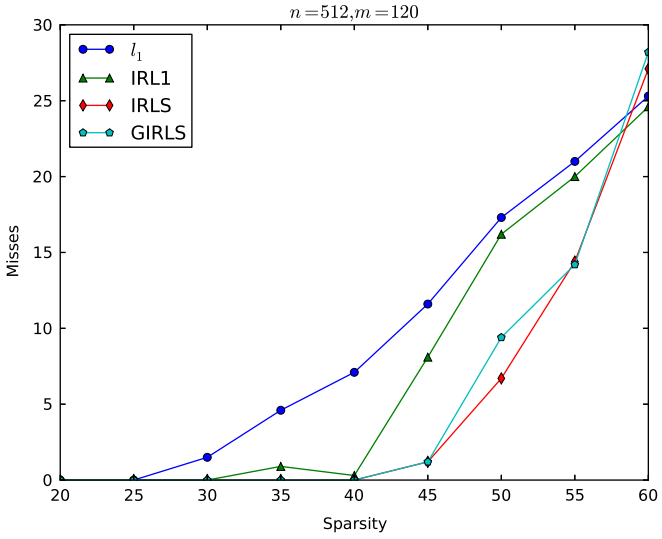


Fig. 7. Misses of ℓ_1 and ℓ_p algorithms (p=0.5).

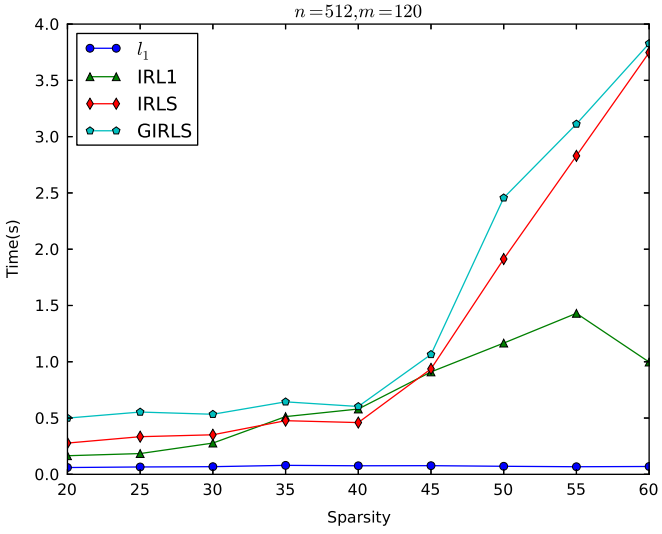


Fig. 8. The running time of ℓ_1 and ℓ_p algorithms (p=0.5).

decreases, the successful recovery rate increases drastically. However, the difference between the successful recovery rates of $p=0.9$ and $p=0.5$ is much larger than that between $p=0.5$ and $p=0.1$. Similar phenomenon has been noticed by Xu et al. [22]. So they advocated $p=0.5$.

We further test with a real application: face recognition with sparse representation proposed by Wright et al. [45]. This method is to solve the following problem:

$$\min_{\mathbf{x}\in\mathbb{R}^n} \lambda\|\mathbf{x}\|_p^p + \frac{1}{2}\|\mathbf{Ry}-\mathbf{RAx}\|_2^2 \tag{32}$$

first, where $\mathbf{y}$ is the test image, $\mathbf{A}=[\mathbf{A}_1,\mathbf{A}_2,\ldots,\mathbf{A}_k]$ is the collection of all training face images, $\mathbf{A}_i=[\mathbf{a}_{i,1},\mathbf{a}_{i,2},\ldots,\mathbf{a}_{i,n_i}]$ is the collection of all training face images of the ith subject, and $\mathbf{R}$ is a Gaussian random matrix for reducing the data dimension, and then classify the test image $\mathbf{y}$ to the class that has the least residual reconstruction error using the training images in that class:

$$\min_i \|\mathbf{y}-\mathbf{A}\delta_i(\mathbf{x}^*)\|_2, \tag{33}$$

where δ_i is an operator that picks out the entries corresponding to the ith subject and sets the rest entries to zeros. To identify invalid test images, Wright et al. [45] further proposed the following Sparsity Concentration Index (SCI):

$$SCI(\mathbf{x}) \doteq \frac{k\cdot(\max_i \|\delta_i(\mathbf{x})\|_1)/\|\mathbf{x}\|_1 - 1}{k-1} \tag{34}$$

to reject a test image as an invalid face image if $SCI(\mathbf{x}^*)<\tau$, where k is the number of groups and $\tau>0$ is a threshold. Obviously, the larger SCI is, the heavier the coefficients concentrate on one subject.

To do real experiments, we use the cropped images in Extended Yale Face Database B [46], which contains 2414 frontal-face images of 38 individuals captured under various lighting conditions, the size of each image being 192×168 pixels. We randomly select 20 images from each subject to form the ith training image submatrix $\mathbf{A}_i$. As we have to do a huge amount of random tests, to facilitate computation we do singular value decomposition (SVD) on each $\mathbf{RA}_i$: $\mathbf{RA}_i=\mathbf{U}_i\Sigma_i\mathbf{V}_i^T$, and replace the matrix $\mathbf{RA}_i$ with the leading five orthonormal faces $\tilde{\mathbf{U}}_i\doteq\mathbf{U}_i(:,1:5)$. Accordingly, the matrix $\mathbf{RA}$ in (32) is replaced with $\mathbf{U}=[\tilde{\mathbf{U}}_1,\tilde{\mathbf{U}}_2,\ldots,\tilde{\mathbf{U}}_{38}]$, which has $n=38\times5=190$ columns. For fair comparison between ℓ_p and ℓ_1, which should use different λ's, we further randomly select another 70 images from each subject and tune the λ's in a range $(10^{-4},10^{-1})$ for ℓ_p and ℓ_1 minimizations, respectively, such that they both achieve the highest recognition rates. We vary the projection dimension (i.e., the number of rows in $\mathbf{R}$) from 20 to 160. For each projection

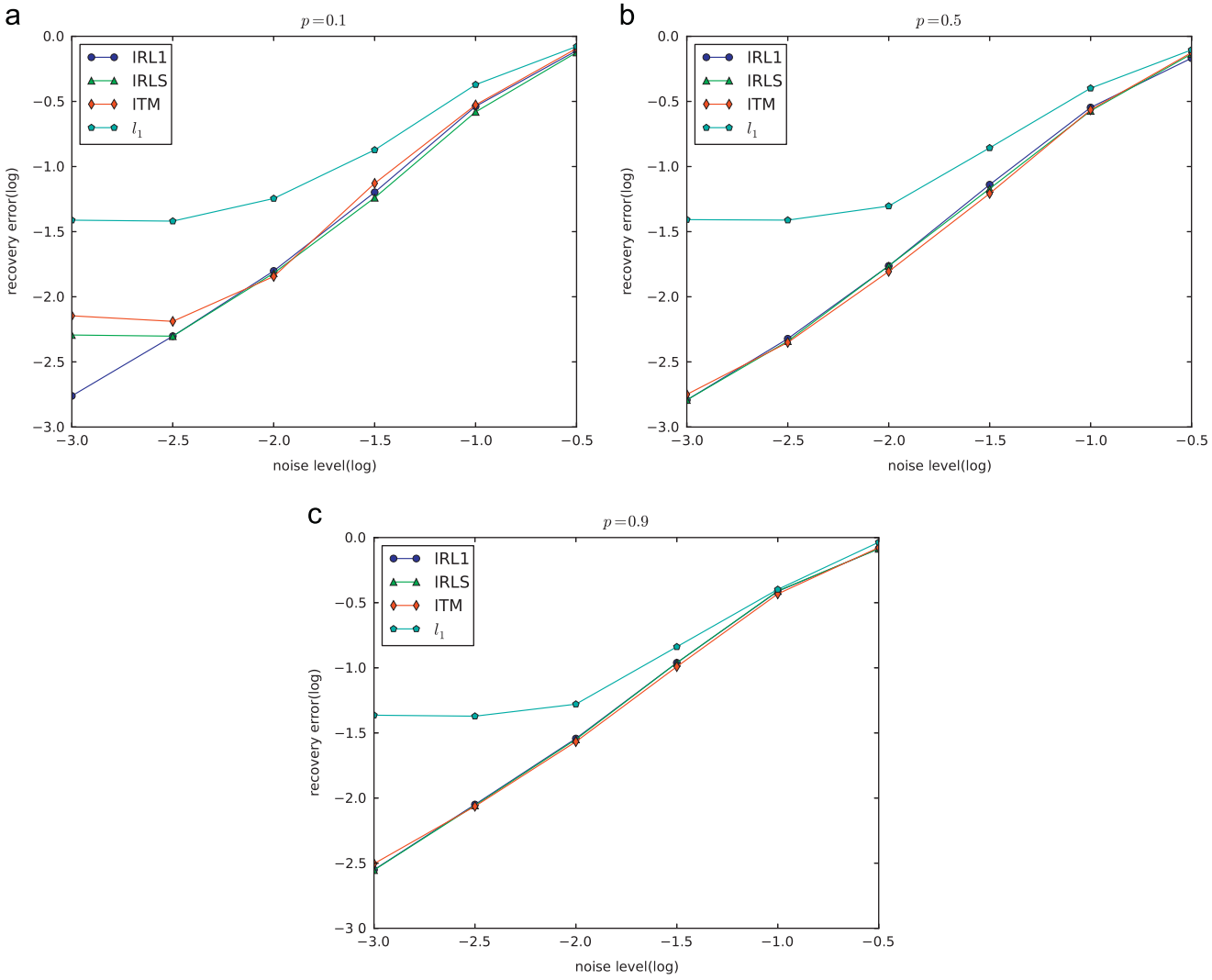


Fig. 9. The recovery errors under different noise levels and p's. (a) $p=0.1$, (b) $p=0.5$, and (c) $p=0.9$.

dimension, we randomly select 500 test images from the remaining images and record the recognition rate and the median of SCI values obtained in the 500 times experiments. The SCI threshold is set to 0.2, and those recoveries with SCI values lower than the threshold will be regarded as mis-classifications as there is no invalid test face images. The value of p is fixed at 0.5. The results are shown in Figs. 2 and 3.

We can see that ℓ_p minimization consistently improves the recognition rates over ℓ_1 minimization by at least 5%. Moreover, ℓ_p minimization consistently has much higher SCI values, which indicates that the nonzero entries of the ℓ_p solutions strongly concentrate on one subject, hence help improve recognition.

To summarize, by solving ℓ_p minimization problems one can get sparser solutions. So ℓ_p minimization is often desirable if sparsity is a critical issue.

3.2. Comparison of algorithms for equality constrained problem

In this experiment, we compare the success rates of recovering the sparsest solution with different algorithms. The value of p is fixed. The algorithms chosen for comparison are IRL1, IRLS, and GIRLS. They are all initialized as $\mathbf{x}^{(0)}=0$ as they all aim at finding the sparsest solution.

We follow Yang et al. [13] to do this experiment. We fix $p=0.5$ and the ambient dimension $n=500$, and calculate for each algorithm the success rates under different sparsity rates $\sigma=S/n\in(0,1]$ and sampling rates $\delta=m/n\in(0,1]$. The test samples $(\mathbf{A},\mathbf{y})$ are generated in the same way as in the last section. Then we apply algorithms to problem Eℓ_p and obtain a recovered vector $\mathbf{x}^*$. $\mathbf{x}^*$ is regarded as a successful recovery if $\|\mathbf{x}^*-\mathbf{x}_0\|_2/\|\mathbf{x}_0\|_2<10^{-3}$. The success rate is obtained by doing 100 times of experiments for each setting of (σ,δ). The 90% success-rate curves are drawn in Fig. 4. We can see that IRLS and GIRLS can recover the sparsest solution better than IRL1, and all ℓ_p algorithms result in much higher success rates than ℓ_1 minimization does. A blow-up plot of the differences from IRL1 (Fig. 5) shows that GIRLS is not much better than IRLS. This is because the nonconvex nature of ℓ_p problem.

Although the relative error $\|\mathbf{x}^*-\mathbf{x}_0\|_2/\|\mathbf{x}_0\|_2$ measures the difference of the computed solution to the ground truth, people may also care about the false alarms and misses in the computed solution. Here false alarm means that an entry in the ground truth is zero but is nonzero in the computed solution; while miss means that an entry is zero in the computed solution but actually it should not be zero. These two measures are important for some applications, e.g., feature selection.

To measure false alarm and miss, we fix $p=0.5$, the ambient dimension to $n=512$ and number of measurements to $m=120$, and test ℓ_p algorithms and ℓ_1 minimization for different sparsity S. For each setting we run the algorithms for 10 times and report the

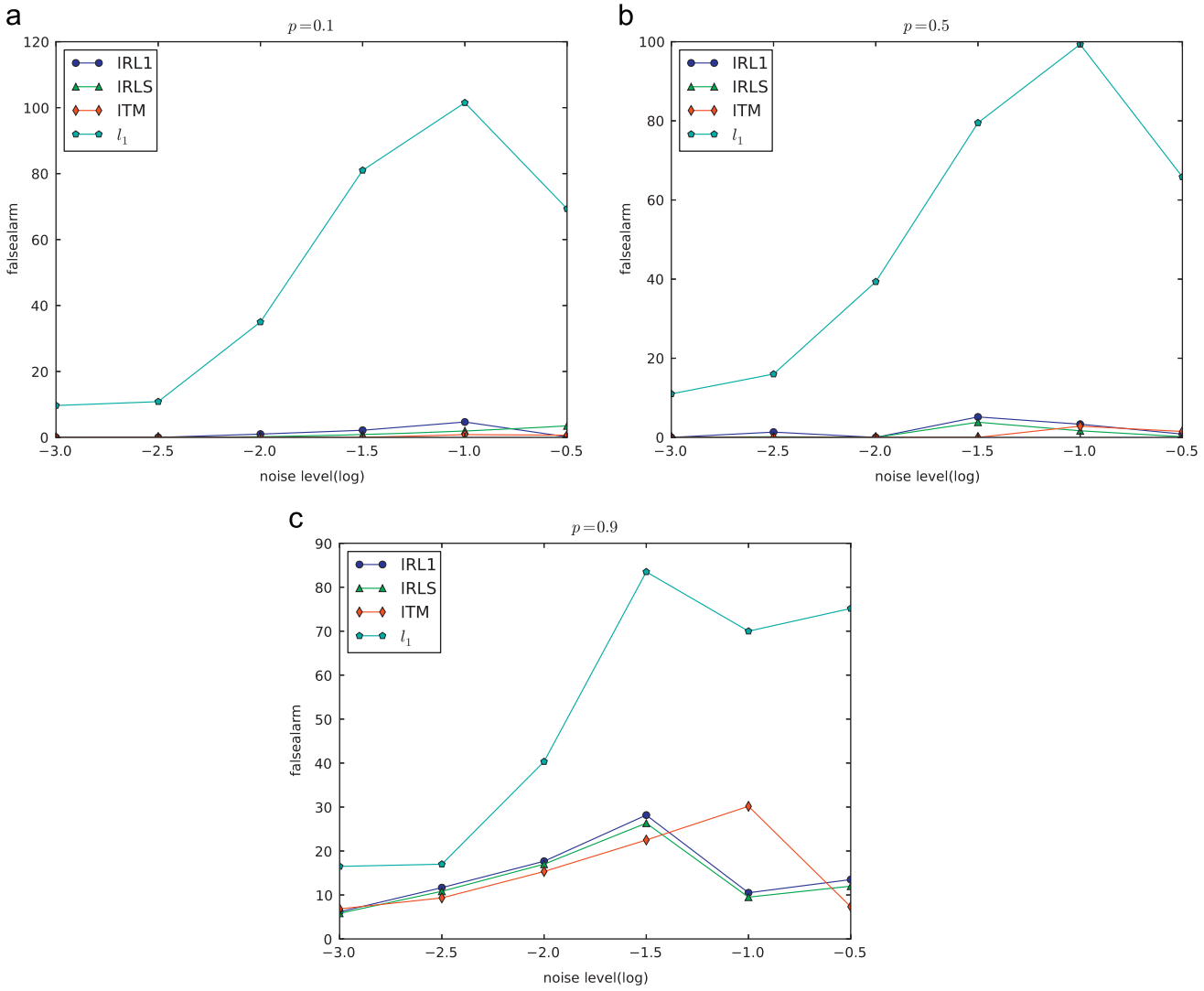


Fig. 10. The number of false alarms under different noise levels and p's. (a) $p=0.1$, (b) $p=0.5$, and (c) $p=0.9$.

average quantity. The results are shown in Figs. 6 and 7. We can see that the false alarms of ℓ_1 minimization are always higher than those of ℓ_p minimization and when $S \leq 55$ the misses of ℓ_1 minimization are also always higher than those of ℓ_p minimization. This is because ℓ_p minimization has higher success rates on recovering the ground truth solution. However, when S is larger the misses of ℓ_p minimization may be nearly the same as, or even be slightly more than, those of ℓ_1 minimization. This is because for larger S, the sparsest solution is less unlikely to be unique and ℓ_p minimization may have found other sparsest solutions. Moreover, it is more likely that 0 is no longer a good initialization. Note that the performances of IRLS and GIRLS are very close to each other and are both better than IRL1.

Finally, we compare the running time of the algorithms. As IRL1 requires solving an ℓ_1 minimization problem at each iteration and IRLS has a closed-form solution at each iteration, it is expected that IRL1 is slower than ℓ_l minimization and IRLS. Moreover, as GIRLS further involves a 1D minimization but usually has fewer iterations, the running time of GIRLS and IRLS should be nearly the same. To verify the above, we use ℓ_1-magic [44] to solve the ℓ_1 minimization and the ℓ_1 subproblems in IRL1. The running time under different sparsities is displayed in Fig. 8, where we fix $p=0.5$, the ambient dimension to $n=512$ and number of measurements to $m=120$. We can see that by using highly optimized packages to solve ℓ_1 problem at each iteration, IRL1 has achieved comparable or even faster speed than IRLS in our experiments. The running time of IRLS and GIRLS is indeed nearly the same as we have not used optimized tools to solve the 1D minimization problem at each iteration of GIRLS. Moreover, IRL1, IRLS and GIRLS are much slower than ℓ_1 minimization because ℓ_1 minimization is a convex program.

By the above experiments and observations, we may have the following conclusions:

1. Measured by success recovery rates, false alarms, and misses, equality constrained ℓ_p minimization indeed can produce sparser solutions than ℓ_1 minimization does.
2. When the ground truth solution is sparse enough, the speed of ℓ_p minimization is acceptable as compared with ℓ_1 minimization.
3. IRLS and GIRLS have very similar performances and are both better than IRL1.
4. By considering both speed and performance, we recommend using IRLS for equality constrained ℓ_p minimization.

3.3. Comparison of algorithms for unconstrained problem

In this section, we compare the performance of different algorithms for the unconstrained problem Lℓ_p. The algorithms

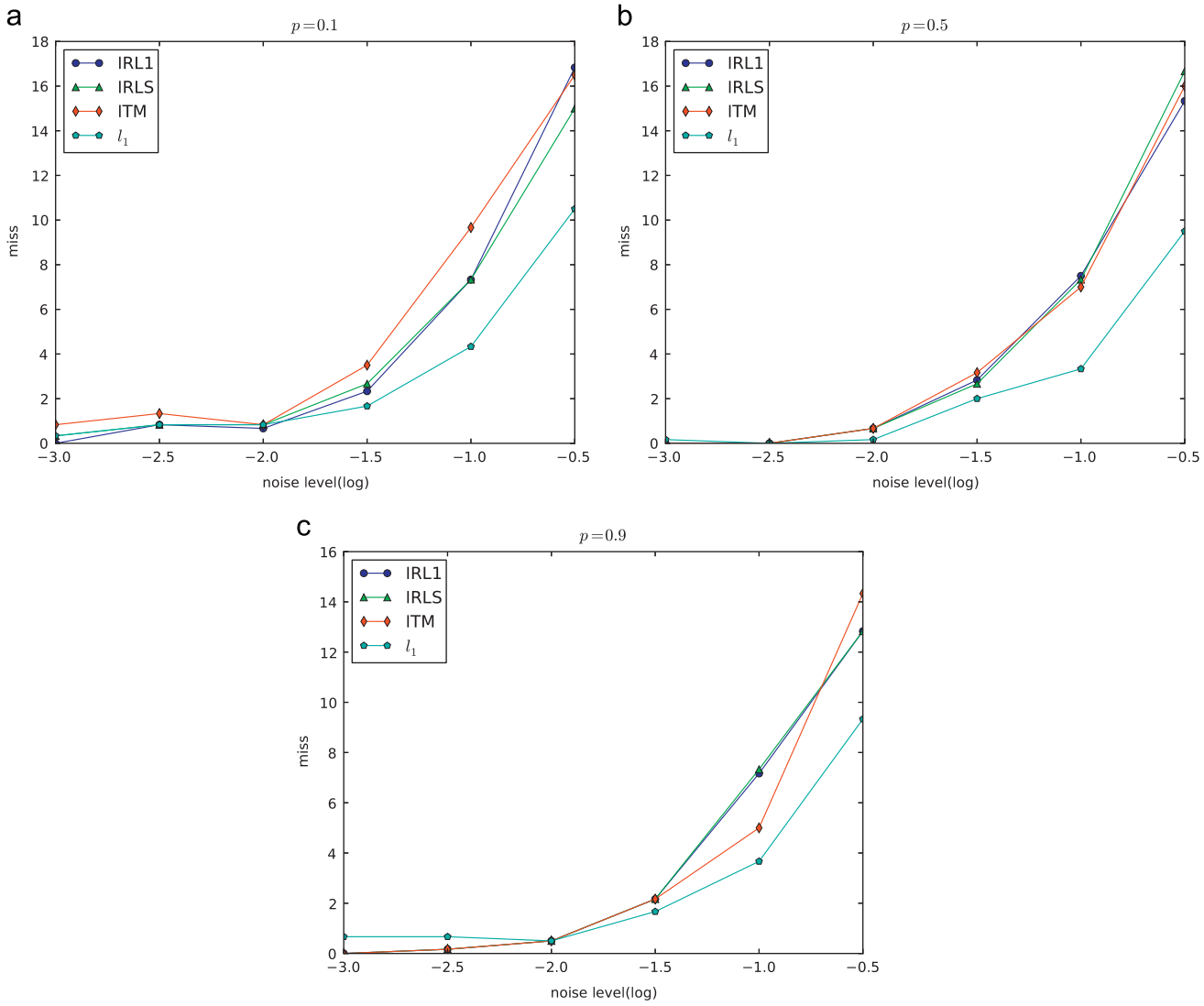


Fig. 11. The number of misses under different noise levels and p's. (a) $p=0.1$, (b) $p=0.5$, and (c) $p=0.9$.

involve IRL1, IRLS, and ITM. We also compare with ℓ_1-norm based unconstrained minimization to show the advantage of ℓ_p over ℓ_1 minimizations.

For fair comparison between ℓ_1 and ℓ_p, which should use different λ's, we tune the λ's in a range $(10^{-5}, 10^{-1})$ for ℓ_p and ℓ_1 minimizations, respectively, such that they both achieve the smallest recovery error. We fix $n=512$, $m=120$, and $S=20$ throughout the experiments.

To prepare test data, we first generate a ground-truth vector $\mathbf{x}_0$ randomly, a Gaussian random matrix $\mathbf{A}$ with normalized columns, and a Gaussian noise $\mathbf{e}$, then the measurement vector $\mathbf{y}$ is obtained by $\mathbf{y}=\mathbf{A}\mathbf{x}_0+\mathbf{e}$. We vary the noise level $\|\mathbf{e}\|_2/\|\mathbf{y}\|_2$ from 10^{-3} to $10^{-0.5}$, and for each setting we test the algorithms for 10 times to obtain the average quantity of recovery error, false alarms, misses, and running time.

The results are shown in Figs. 9–12. We can have the following observations:

1. ℓ_p minimization recovers the ground truth much better than ℓ_1 minimization does, especially when the noise level is low and p is small. Note that the misses of ℓ_p and ℓ_1 minimizations are nearly the same, while ℓ_p minimization results in much fewer false alarms for high noise level case, thus ℓ_p minimization achieves much sparser solutions than ℓ_1 does when there is considerable amount of noise. However, ℓ_1 minimization is again much faster than ℓ_p minimization as expected.
2. ℓ_p algorithms have nearly the same performance, they only have significant difference on speed. ITM is the fastest among ℓ_p algorithms, and its performance is competitive. Note that ITM has obtained a whole solution path, thus it is much faster than IRL1 and IRLS when solutions under different parameters are needed.
3. There is not much difference between $p=0.1$ and $p=0.5$ for the performance of ℓ_p algorithms. When p approaches 1, say, $p=0.9$, the speed and false alarms of ℓ_p algorithms becomes much worse. So we suggest $p=0.5$ for practical use. This is also consistent with [22].

So we may conclude that ℓ_p minimization leads to sparser solutions than ℓ_1 does by sacrificing efficiency. Considering both performance and speed, we recommend IRLS again for solving the unconstrained ℓ_p minimization problem when the parameter is given. To obtain solutions under different λ's, e.g., when tuning parameters, we recommend ITM instead.

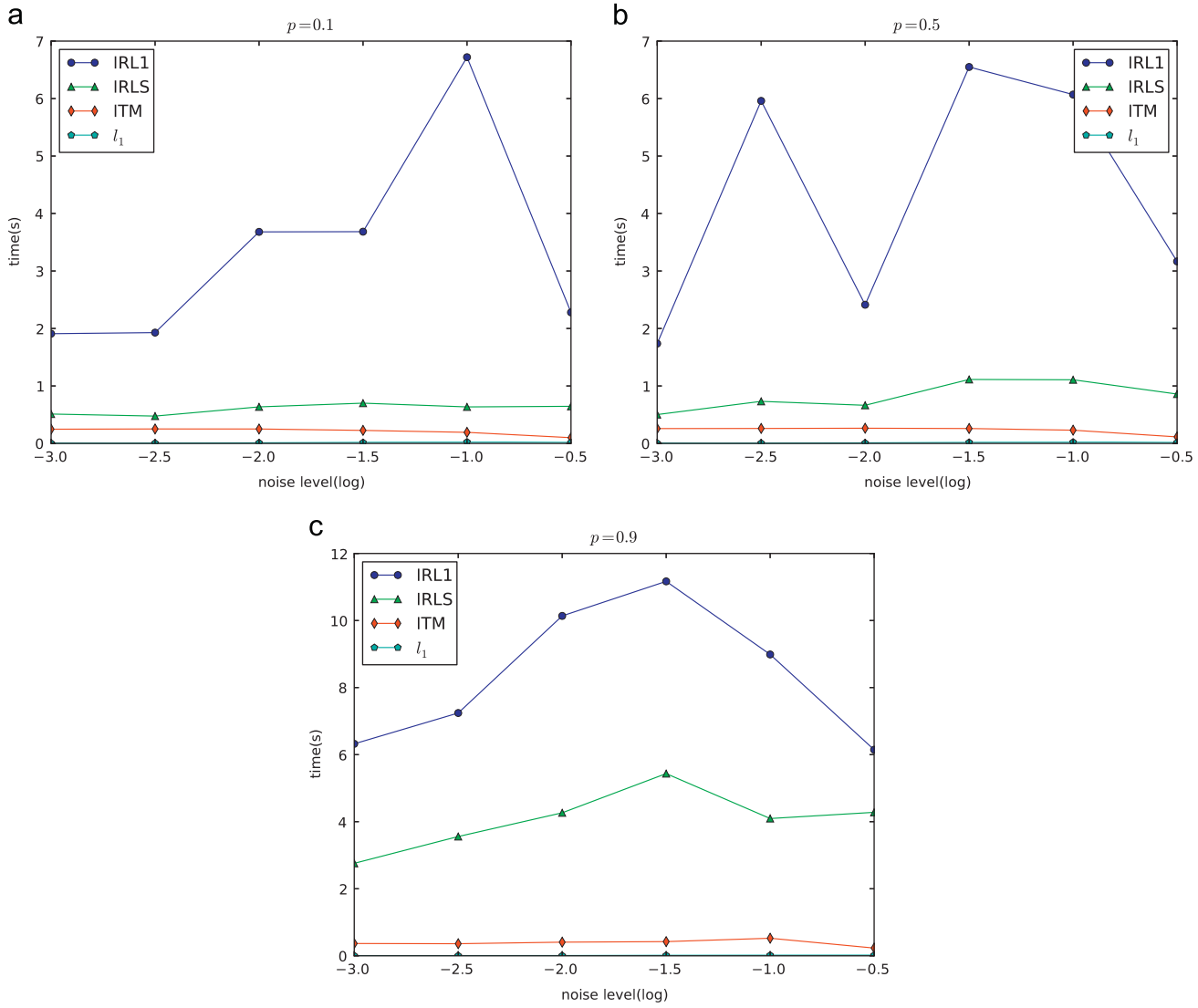


Fig. 12. The running times under different noise levels and p's. (a) $p=0.1$, (b) $p=0.5$ and (c) $p=0.9$.

4. Conclusions and remarks

In this paper, we have reviewed three typical ℓ_p minimization algorithms: IRL1, IRLS (and its generalized form GIRLS), and ITM, and compared their performance on equality constrained ℓ_p minimization problem and unconstrained ℓ_p minimization problem. We have found that IRLS is the best among the three algorithms, regarding performance and computation speed. We also compare ℓ_p minimization with ℓ_1 minimization and verify that ℓ_p minimization can result in sparser solution than ℓ_1 minimization does, justifying the necessity of using ℓ_p minimization for solving the sparsest representation vector. As ℓ_p minimization is generally slower than ℓ_1 minimization, in real applications we recommend using ℓ_p minimization (and solving by IRLS) only when the sparsity of solution is a critical issue.

Finally, we would like to mention some practical issues when using these algorithms. First, although we recommend IRLS, as mentioned in Section 2.2.2, one has to choose an appropriate threshold to enforce the sparsity of its solution. Otherwise, false alarms or misses may occur. In comparison, IRL1 and ITM do not have such an issue as they both contain thresholding operations in their iterations. Second, in practice people may want to obtain a whole solution path, i.e., solutions under different λ's, for parameter tuning, where λ starts from a relatively large value and gradually reduces to a relatively small value. In this case, ITM is more preferred over IRLS because the *warm start* technique can be easily incorporated in ITM to boost its speed significantly, as having been showed in Fig. 12. The warm start technique simply uses the solution of last problem as the initial value of next problem. Third, as for the initialization of the solution for the unconstrained ℓ_p problem, no theoretical analysis has been developed for IRL1 and IRLS. So in practice one may simply initialize with a zero vector. However, for ITM She has reported in [23] that if starting with the zero vector ITM with warm start can easily be trapped at a poor local optimum, hence initializing with a zero vector is not recommended for ITM. Nonetheless, how to choose a good initial vector is still an open problem.

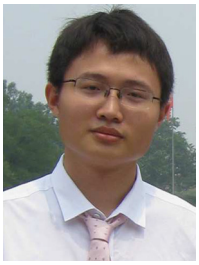
Acknowledgments

The authors are grateful to the associate editor and the two anonymous referees for their careful comments and useful suggestions. Qin Lyu and Chao Zhang are supported by National Key Basic Research Project of China (973 Program) 2011CB302400 and National Nature Science Foundation of China (NSFC Grant, no. 61071156). Zhouchen Lin is supported by NSFC Grants (Nos. 61272341, 61231002 and 61121002). Yiyuan She is partially supported by NSF Grant CCF-1116447.

References

[1] D.L. Donoho, Compressed sensing, IEEE Trans. Inf. Theory 52 (4) (2006) 1289–1306.
[2] E.J. Candès, J.K. Romberg, T. Tao, Stable signal recovery from incomplete and inaccurate measurements, Commun. Pure Appl. Math. 59 (8) (2006) 1207–1223.
[3] B.K. Natarajan, Sparse approximate solutions to linear systems, SIAM J. Comput. 24 (2) (1995) 227–234.
[4] D.L. Donoho, For most large underdetermined systems of linear equations the minimal ℓ_1-norm solution is also the sparsest solution, Commun. Pure Appl. Math. 59 (7) (2006) 907–934.
[5] G.M. Davis, S. Mallat, M. Avellaneda, Adaptive greedy approximations, J. Constructive Approximation 13 (1) (1997) 57–98.
[6] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, Least angle regression, Ann. Stat. 32 (2) (2004) 407–499.
[7] M.A. Figueiredo, R.D. Nowak, S.J. Wright, Gradient projection for sparse reconstruction: application to compressed sensing and other inverse problems, IEEE J. Sel. Top. Signal Process. 1 (4) (2007) 586–597.
[8] D.M. Malioutov, A.S. Willsky, Homotopy continuation for sparse signal representation, in: IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 5, 2005, pp. 733–736.
[9] P.L. Combettes, V.R. Wajs, Signal recovery by proximal forward-backward splitting, Multiscale Modeling Simulation 4 (4) (2005) 1168–1200.
[10] A. Beck, M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems, SIAM J. Imaging Sci. 2 (1) (2009) 183–202.
[11] J. Yang, Y. Zhang, Alternating direction algorithms for ℓ_1-problems in compressive sensing, SIAM J. Sci. Comput. 33 (1) (2011) 250–278.
[12] J. Yang, X. Yuan, Linearized augmented Lagrangian and alternating direction methods for nuclear norm minimization, Math. Comput. 82 (281) (2013) 301–329.
[13] A. Yang, A. Ganesh, S. Sastry, Y. Ma, Fast ℓ_1-minimization algorithms and an application in robust face recognition: a review, in: IEEE International Conference on Image Processing, 2010, pp. 1849–1852.
[14] M. Schmidt, G. Fung, R. Rosales, Fast optimization methods for ℓ_1 regularization: a comparative study and two new approaches, Eur. Conf. Mach. Learn. 4701 (2007) 286–297.
[15] M. Schmidt, G. Fung, R. Rosales, Optimization methods for ℓ_1-regularization, Technical Report, University of British Columbia, 2009.
[16] R. Saab, R. Chartrand, O. Yilmaz, Stable sparse approximations via nonconvex optimization, in: IEEE International Conference on Acoustics, Speech and Signal Processing, 2008, pp. 3885–3888.
[17] R. Chartrand, Nonconvex compressed sensing and error correction, in: IEEE International Conference on Acoustics, Speech and Signal Processing, vol. 3, 2007, pp. 889–892.
[18] T. Zhang, Multi-stage convex relaxation for learning with sparse regularization, Adv. Neural Inf. Process. Syst. 21 (2009) 1929–1936.
[19] S. Foucart, M. Lai, Sparsest solutions of underdetermined linear systems via ℓ_q-minimization for $0 < q < 1$, Appl. Comput. Harmonic Anal. 26 (3) (2009) 395–407.
[20] G. Gasso, A. Rakotomamonjy, S. Canu, Recovering sparse signals with a certain family of nonconvex penalties and DC programming, IEEE Trans. Signal Process. 57 (12) (2009) 4686–4698.
[21] M. Lai, J. Wang, An unconstrained $_q$ minimization with $0 < q < 1$ for sparse solution of under-determined linear systems, SIAM J. Optim. 21 (1) (2011) 82–101.
[22] Z. Xu, X. Chang, F. Xu, H. Zhang, $_{1/2}$ regularization: a thresholding representation theory and a fast solver, IEEE Trans. Neural Networks Learn. Syst. 23 (7) (2012) 1013–1027.
[23] Y. She, An iterative algorithm for fitting nonconvex penalized generalized linear models with grouped predictors, Comput. Stat. Data Anal. 9 (2012) 2976–2990.
[24] N. Mourad, J.P. Reilly, $_p$ minimization for sparse vector reconstruction, in: IEEE International Conference on Acoustics, Speech, and Signal Processing, 2009, pp. 3345–3348.
[25] N. Mourad, J.P. Reilly, Minimizing nonconvex functions for sparse vector reconstruction, IEEE Trans. Signal Process. 58 (7) (2010) 3485–3496.
[26] R. Chartrand, W. Yin, Iteratively reweighted algorithms for compressive sensing, in: IEEE International Conference on Acoustics, Speech, and Signal Processing, 2008, pp. 3869–3872.
[27] R. Chartrand, Exact reconstruction of sparse signals via nonconvex minimization, IEEE Signal Process. Lett. 14 (10) (2007) 707–710.
[28] H. Zou, R. Li, One-step sparse estimates in nonconcave penalized likelihood models, Ann. Stat. 36 (4) (2008) 1509–1533.
[29] E.J. Candès, M. Wakin, S. Boyd, Enhancing sparsity by reweighted ℓ_1 minimization, J. Fourier Anal. Appl. 14 (5) (2008) 877–905.
[30] X. Chen, W. Zhou, Convergence of reweighted $_1$ minimization algorithms and unique solution of truncated ℓ_p minimization, Technical Report, Hong Kong Polytechnic University, 2010.
[31] R. Horst, N.V. Thoai, DC programming: overview, J. Optim. Theory Appl. 103 (1) (1999) 1–43.
[32] B. Sriperumbudur, G. Lanckriet, On the convergence of the concave–convex procedure, Adv. Neural Inf. Process. Syst. 22 (2009) 1759–1767.
[33] H. Zou, The adaptive LASSO and its oracle properties, J. Am. Stat. Assoc. 101 (476) (2006) 1418–1429.
[34] B.D. Rao, K. Kreutz-Delgado, An affine scaling methodology for best basis selection, IEEE Trans. Signal Process. 47 (1) (1999) 187–200.
[35] I.F. Gorodnitsky, B.D. Rao, A recursive weighted minimum norm algorithm: analysis and applications, in: IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 3, 1993, pp. 456–459.
[36] I.F. Gorodnitsky, B.D. Rao, Sparse signal reconstruction from limited data using FOCUSS: a re-weighted minimum norm algorithm, IEEE Trans. Signal Process. 45 (3) (1997) 600–616.
[37] R. Chartrand, V. Staneva, Restricted isometry properties and nonconvex compressive sensing, Inverse Probl. 24 (2008) 035020.
[38] J. Fan, R. Li, Variable selection via nonconcave penalized likelihood and its oracle properties, J. Am. Stat. Assoc. 96 (456) (2001) 1348–1360.
[39] D.R. Hunter, R. Li, Variable selection using MM algorithms, Ann. Stat. 33 (4) (2005) 1617.
[40] M. Lai, Y. Xu, W. Yin, Improved iteratively reweighted least squares for unconstrained smoothed ℓ_1-regularization, SIAM J. Numer. Anal. 51 (2013) 927–957, http://dx.doi.org/10.1137/110840364.
[41] Y. She, Thresholding-based iterative selection procedures for model selection and shrinkage, Electron. J. Stat. 3 (2009) 384–415.
[42] J. Friedman, T. Hastie, H. Höfling, R. Tibshirani, Pathwise coordinate optimization, Ann. Appl. Stat. 1 (2) (2007) 302–332.
[43] J. Friedman, T. Hastie, R. Tibshirani, Regularization paths for generalized linear models via coordinate descent, J. Stat. Software 33 (1) (2010) 1–22.

[44] E.J. Candès, J.K. Romberg, ℓ_1-MAGIC toolbox.
[45] J. Wright, A.Y. Yang, A. Ganesh, S. Sastry, Y. Ma, Robust face recognition via sparse representation, IEEE Trans. Pattern Anal. Mach. Intell. 31 (2) (2009) 210–227.
[46] A.S. Georghiades, P.N. Belhumeur, D.J. Kriegman, From few to many: illumination cone models for face recognition under variable lighting and pose, IEEE Trans. Pattern Anal. Mach. Intell. 23 (6) (2001) 643–660.



Qin Lyu received dual bachelors' degrees in automation and mathematics from Tsinghua University, Beijing, China, in 2011. He is currently a master candidate at Key Laboratory of Machine Perception (MOE), School of Electronics Engineering and Computer Science, Peking University. His research interests include statistical learning, numerical optimization, and computer vision.

Zhouchen Lin received the Ph.D. degree in applied mathematics from Peking University in 2000. He is currently a Professor at Key Laboratory of Machine Perception (MOE), School of Electronics Engineering and Computer Science, Peking University. He is also a Chair Professor at Northeast Normal University and a guest professor at Beijing Jiaotong University. Before March 2012, he was a Lead Researcher at Visual Computing Group, Microsoft Research Asia. He was a guest professor at Shanghai Jiaotong University and Southeast University, and a guest researcher at Institute of Computing Technology, Chinese Academy of Sciences. His research interests include computer vision, image processing, computer graphics, machine learning, pattern recognition, and numerical computation and optimization. He is an associate editor of International J. Computer Vision and Neurocomputing and a senior member of the IEEE.

Yiyuan She obtained his Ph.D. degree in Statistics at Stanford University in 2008. He is currently an Assistant Professor in the Department of Statistics at Florida State University. His research interests include high-dimensional model selection, reduced rank models, robust statistics, statistics computing, and bioinformatics.

Chao Zhang received the Ph.D. degree in electrical engineering from Northern Jiaotong University, Beijing, China in 1995. After working as a Postdoctoral Research Fellow for two years, he became a faculty member in June 1997 at the National Laboratory on Machine Perception, Peking University, where he is currently an Associate Professor. His research interests include computer vision, statistical pattern recognition, and video-based biometrics.