Lp Quasi-norm Minimization

M.E. Ashour*, C.M. Lagoa* and N.S. Aybat†
* The Department of Electrical Engineering and Computer Science, The Pennsylvania State University.
† The Department of Industrial Engineering, The Pennsylvania State University.

***Abstract*—The ℓ_p $(0 < p < 1)$ quasi-norm is used as a sparsity-inducing function, and has applications in diverse areas, e.g., statistics, machine learning, and signal processing. This paper proposes a heuristic based on a two-block ADMM algorithm for tackling ℓ_p quasi-norm minimization problems. For $p = s/q < 1$, $s, q \in \mathbb{Z}_+$, the proposed algorithm requires solving for the roots of a scalar degree $2q$ polynomial as opposed to applying a soft thresholding operator in the case of ℓ_1. We show numerical results for two example applications, sparse signal reconstruction from few noisy measurements and spam email classification using support vector machines. Our method obtains significantly sparser solutions than those obtained by ℓ_1 minimization while achieving similar level of measurement fitting in signal reconstruction, and training and test set accuracy in classification.**

I. Introduction

This paper considers problems of the form:

$$\min_{\mathbf{x}} \quad \|\mathbf{x}\|_p^p \triangleq \textstyle\sum_{i\in[n]} |x_i|^p \quad \text{s.t.} \quad f(\mathbf{x}) \leq 0, \qquad (1)$$

where $p \in (0, 1)$, $[n] = \{1, \ldots, n\}$, $n \in \mathbb{Z}_+$, and $f : \mathbb{R}^n \to \mathbb{R}$ is a convex possibly nonsmooth function. This formulation arises in areas as machine learning and signal processing. For instance, let $\{(\mathbf{u}_i, v_i)\}_{i\in[m]}$ be a training set of feature-label pairs $(\mathbf{u}_i, v_i)$, and $m \in \mathbb{Z}_+$. In regression, one seeks to fit a model that relates $v_i \in \mathbb{R}$ to $\mathbf{u}_i$ via solving (1) with $f(\mathbf{x}) = \|\mathbf{U}\mathbf{x} - \mathbf{v}\|^2 - \epsilon$, where the ith row of $\mathbf{U} \in \mathbb{R}^{m\times n}$ is constructed using $\mathbf{u}_i$, $\mathbf{v} = [v_i]_{i\in[m]}$, and $\epsilon > 0$. Alternatively, if $v_i \in \{-1, 1\}$ denotes a class label in a binary classification problem, one might seek to find a linear classifier with a decision rule $\hat{v} = \text{sign}(\mathbf{u}^\top \mathbf{x})$, e.g., using support vector machines, where the first entry of $\mathbf{u}$ and $\mathbf{x}$ are 1 and the bias term, respectively. The classifier can be obtained by solving (1) with $f(\mathbf{x}) = \frac{1}{m}\sum_{i\in[m]} \left(1 - v_i \mathbf{u}_i^\top \mathbf{x}\right)^+ - \epsilon$, where $(.)^+ = \max(., 0)$. In both examples, the ℓ_p quasi-norm of $\mathbf{x}$ is used in (1) as a sparsity-inducing function. Problem (1) provides a tradeoff between how well a model performs on a certain task versus its complexity, controlled by ϵ.

We propose an ADMM algorithm for approximating the solution of (1). For $p = s/q < 1$, the computational complexity of the proposed algorithm is similar to ℓ_1 minimization except for the additional effort of solving for the roots of a scalar degree $2q$ polynomial as opposed to applying the soft thresholding operator for ℓ_1. We present numerical results showing that our method significantly outperforms ℓ_1 minimization in terms of the sparsity level of obtained solutions.

This work was partially supported by National Institutes of Health (NIH) Grant R01 HL142732, National Science Foundation (NSF) Grant 1808266.

A sparse solution to (1) is defined as one that has a small number of entries whose magnitudes are significantly different than zero [1]. Indeed, many signals/images are either sparse or compressible, i.e., can be approximated by a sparse representation with respect to some transform domain. The development of a plethora of overcomplete waveform dictionaries motivate the basis pursuit principle that decomposes a signal into a sparse superposition of dictionary elements [2]. Furthermore, sparsity finds application in object recognition and classification problems, e.g., [5], and signal estimation from incomplete linear measurements known as compressed sensing [6], [7]. Reference [8] provides a comprehensive review of theoretical results on sparse solutions of linear systems and its applications in inverse problems.

Retrieving sparse solutions of underdetermined linear systems received tremendous attention over the past two decades; see [8] and references therein. Reference [10] identifies the major algorithmic approaches for tackling sparse approximation problems, namely, greedy pursuit [11], convex relaxation [2], [6], [7], [12], and nonconvex optimization [13], [14].

Problems seeking sparse solutions are often posed as $\min\{f(\mathbf{x}) + \mu g(\mathbf{x})\}$ for some $\mu > 0$ and a sparsity-inducing penalty function g, e.g., $g(\mathbf{x}) = \|\mathbf{x}\|_p^p$, where g can be either convex, e.g., $p = 1$, or nonconvex, e.g., $0 \leq p < 1$. For a comprehensive reference on sparsity-inducing penalty functions, see [15]. It has been shown that exact sparse signal recovery from few measurements is possible via letting g be the ℓ_1 norm if the measurement matrix satisfies a certain restricted isometry property (RIP) [1], [16]. However, RIP is a stringent property. Motivated by the fact that $\|\mathbf{x}\|_p^p \to \|\mathbf{x}\|_0$ as $p \to 0$, it is natural to consider the ℓ_p quasi-norm problem $(0 < p < 1)$. It has been shown in [17] that ℓ_p minimization with $p < 1$ achieves perfect signal reconstruction under less restrictive isometry conditions than needed for ℓ_1. Several references considered sparse signal reconstruction via nonconvex optimization, [13], [17]–[21] to name a few. In [13], it is shown that replacing ℓ_1 norm with ℓ_p quasi-norm, signal recovery is possible using fewer measurements. Furthermore, [13] presents a simple projected gradient descent method that identifies a local minimizer of the problem. An algorithm that uses operator splitting and Bregman iteration methods as well as a shrinkage operator is presented in [18]. Reference [19] proposes an algorithm based on the idea of locally replacing the original nonconvex objective function by quadratic convex functions that are easily minimized and establishes connection to iterative reweighted least squares [22]. In [20], an interior point potential reduction algorithm

is proposed to compute an ϵ-KKT solution in $\mathcal{O}(\frac{n}{\epsilon}\log\frac{1}{\epsilon})$ iterations, where n is the dimension of $\mathbf{x}$. Reference [21] uses ADMM and proposes a generalized shrinkage operator for nonconvex sparsity-inducing penalty functions.

II. Algorithm

This section develops a method for approximating the solution of (1). Problem (1) is convex at $p = 1$; hence, can be solved efficiently and exactly. However, the problem becomes nonconvex when $p < 1$. An epigraph equivalent formulation of (1) is obtained by introducing the variable $\mathbf{t} = [t_i]_{i\in[n]}$:

$$\begin{array}{ll}\min\limits_{\mathbf{x},\mathbf{t}} & \mathbf{1}^\top\mathbf{t} \\ \text{s.t.} & t_i \geq |x_i|^p, \quad i \in [n] \\ & f(\mathbf{x}) \leq 0,\end{array} \tag{2}$$

where $\mathbf{1}$ is a vector of all ones. Let the nonconvex set $\mathcal{X} \subset \mathbb{R}^2$ be the epigraph of the scalar function $|x|^p$, i.e., $\mathcal{X} = \{(x,t) \in \mathbb{R}^2 : t \geq |x|^p\}$. Then, (2) can be cast as

$$\begin{array}{ll}\min\limits_{\mathbf{x},\mathbf{t}} & \sum\limits_{i\in[n]} \mathbb{1}_{\mathcal{X}}(x_i,t_i) + \mathbf{1}^\top\mathbf{t} \\ \text{s.t.} & f(\mathbf{x}) \leq 0,\end{array} \tag{3}$$

where $\mathbb{1}_{\mathcal{X}}(.)$ is the indicator function to the set $\mathcal{X}$, i.e., it evaluates to zero if its argument belongs to the set $\mathcal{X}$ and is $+\infty$ otherwise. ADMM exploits the structure of the problem to split the optimization over the variables via iteratively solving fairly simple subproblems. In particular, we introduce auxiliary variables $\mathbf{y} = [y_i]_{i\in[n]}$ and $\mathbf{z} = [z_i]_{i\in[n]}$ and obtain an ADMM equivalent formulation of (3) given by:

$$\begin{array}{lll}\min\limits_{\mathbf{x},\mathbf{t},\mathbf{y},\mathbf{z}} & \sum\limits_{i\in[n]} \mathbb{1}_{\mathcal{X}}(x_i,t_i) + \mathbb{1}_{\mathcal{Y}}(\mathbf{y}) + \mathbf{1}^\top\mathbf{z} & \\ \text{s.t.} & \mathbf{x} = \mathbf{y}: & \boldsymbol{\lambda} \\ & \mathbf{t} = \mathbf{z}: & \boldsymbol{\theta},\end{array} \tag{4}$$

where $\mathcal{Y}$ is the 0-sublevel set of f, i.e., $\mathcal{Y} = \{\mathbf{y} \in \mathbb{R}^n : f(\mathbf{y}) \leq 0\}$. The dual variables associated with the constraints $\mathbf{x} = \mathbf{y}$ and $\mathbf{t} = \mathbf{z}$ are $\boldsymbol{\lambda}$ and $\boldsymbol{\theta}$, respectively. Hence, the Lagrangian function corresponding to (4) augmented with a quadratic penalty on the violation of the equality constraints, with penalty parameter $\rho > 0$, is given by:

$$\begin{aligned}L_\rho(\mathbf{x},\mathbf{t},\mathbf{y},\mathbf{z},\boldsymbol{\lambda},\boldsymbol{\theta}) &= \sum_{i\in[n]} \mathbb{1}_{\mathcal{X}}(x_i,t_i) + \mathbb{1}_{\mathcal{Y}}(\mathbf{y}) + \mathbf{1}^\top\mathbf{z} \\ &+ \boldsymbol{\lambda}^\top(\mathbf{x}-\mathbf{y}) + \boldsymbol{\theta}^\top(\mathbf{t}-\mathbf{z}) + \frac{\rho}{2}\big(\|\mathbf{x}-\mathbf{y}\|^2 + \|\mathbf{t}-\mathbf{z}\|^2\big). \end{aligned} \tag{5}$$

Consider the two block variables $(\mathbf{x},\mathbf{t})$ and $(\mathbf{y},\mathbf{z})$. Then, ADMM [23] consists of the following iterations:

$$(\mathbf{x},\mathbf{t})^{k+1} = \operatorname*{argmin}_{\mathbf{x},\mathbf{t}} L_\rho(\mathbf{x},\mathbf{t},\mathbf{y}^k,\mathbf{z}^k,\boldsymbol{\lambda}^k,\boldsymbol{\theta}^k) \tag{6}$$

$$(\mathbf{y},\mathbf{z})^{k+1} = \operatorname*{argmin}_{\mathbf{y},\mathbf{z}} L_\rho(\mathbf{x}^{k+1},\mathbf{t}^{k+1},\mathbf{y},\mathbf{z},\boldsymbol{\lambda}^k,\boldsymbol{\theta}^k) \tag{7}$$

$$\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \rho(\mathbf{x}^{k+1} - \mathbf{y}^{k+1}) \tag{8}$$

$$\boldsymbol{\theta}^{k+1} = \boldsymbol{\theta}^k + \rho(\mathbf{t}^{k+1} - \mathbf{z}^{k+1}). \tag{9}$$

Algorithm 1: ADMM ($\rho > 0$)

Initialize: $\mathbf{y}^0$, $\mathbf{z}^0$, $\boldsymbol{\lambda}^0$, $\boldsymbol{\theta}^0$
for $k \geq 0$ **do**
  $(x_i,t_i)^{k+1} \leftarrow \Pi_{\mathcal{X}}\left(y_i^k - \frac{\lambda_i^k}{\rho}, z_i^k - \frac{\theta_i^k}{\rho}\right), \forall i \in [n]$
  $\mathbf{y}^{k+1} \leftarrow \Pi_{\mathcal{Y}}\left(\mathbf{x}^{k+1} + \frac{\boldsymbol{\lambda}^k}{\rho}\right)$
  $\mathbf{z}^{k+1} \leftarrow \mathbf{t}^{k+1} + \frac{\boldsymbol{\theta}^k - \mathbf{1}}{\rho}$
  $\boldsymbol{\lambda}^{k+1} \leftarrow \boldsymbol{\lambda}^k + \rho(\mathbf{x}^{k+1} - \mathbf{y}^{k+1})$
  $\boldsymbol{\theta}^{k+1} \leftarrow \boldsymbol{\theta}^k + \rho(\mathbf{t}^{k+1} - \mathbf{z}^{k+1})$.

According to the expression of the augmented Lagrangian function in (5), it follows from (6) that the variables $\mathbf{x}$ and $\mathbf{t}$ are updated via solving the following nonconvex problem:

$$\begin{array}{ll}\min\limits_{\mathbf{x},\mathbf{t}} & \|\mathbf{x} - \mathbf{y}^k + \frac{\boldsymbol{\lambda}^k}{\rho}\|^2 + \|\mathbf{t} - \mathbf{z}^k + \frac{\boldsymbol{\theta}^k}{\rho}\|^2 \\ \text{s.t.} & (x_i,t_i) \in \mathcal{X}, \quad i \in [n].\end{array} \tag{10}$$

Exploiting the separable structure of (10), one immediately concludes that (10) splits into n independent 2-dimensional problems that can be solved in parallel, i.e., for each $i \in [n]$,

$$(x_i,t_i)^{k+1} = \Pi_{\mathcal{X}}\left(y_i^k - \frac{\lambda_i^k}{\rho}, z_i^k - \frac{\theta_i^k}{\rho}\right), \tag{11}$$

where $\Pi_{\mathcal{X}}(.)$ denotes the Euclidean projection operator onto the set $\mathcal{X}$. Furthermore, (5) and (7) imply that $\mathbf{y}$ and $\mathbf{z}$ are independently updated as follows:

$$\mathbf{y}^{k+1} = \Pi_{\mathcal{Y}}\left(\mathbf{x}^{k+1} + \frac{\boldsymbol{\lambda}^k}{\rho}\right) \tag{12}$$

$$\mathbf{z}^{k+1} = \mathbf{t}^{k+1} + \frac{\boldsymbol{\theta}^k - \mathbf{1}}{\rho}. \tag{13}$$

Algorithm 1 summarizes the proposed ADMM algorithm. It is clear that $\mathbf{z}$, $\boldsymbol{\lambda}$, and $\boldsymbol{\theta}$ merit closed form updates. However, updating $(\mathbf{x},\mathbf{t})$ requires solving n nonconvex problems. Moreover, a projection onto the convex set $\mathcal{Y}$ is needed for updating $\mathbf{y}$ which can lead to a heavy computational burden. In the following two sections, we present our approach for handling these two concerns.

III. Nonconvex Projection

In this section, we present the method used to tackle the nonconvex projection problem required to update $\mathbf{x}$ and $\mathbf{t}$.

Among the advantages of the proposed algorithm is that it is amenable to decentralization. As is clear from (11), $\mathbf{x}$ and $\mathbf{t}$ can be updated element-wise via performing a projection operation onto the nonconvex set $\mathcal{X}$, one for each $i \in [n]$. The n projection problems can be run independently in parallel. We now outline the proposed idea for solving one such projection, i.e., we suppress the dependence on the index of the entry of $\mathbf{x}$ and $\mathbf{t}$. For $(\bar{x},\bar{t}) \in \mathbb{R}^2$, $\Pi_{\mathcal{X}}(\bar{x},\bar{t})$ entails solving

$$\begin{array}{ll}\min\limits_{x,t} & g(x,t) \triangleq (t-\bar{t})^2 + (x-\bar{x})^2 \\ \text{s.t.} & t \geq |x|^p.\end{array} \tag{14}$$

If $\bar{t} \geq |\bar{x}|^p$, then trivially $\Pi_{\mathcal{X}}(\bar{x}, \bar{t}) = (\bar{x}, \bar{t})$. Thus, we focus on the case in which $\bar{t} < |\bar{x}|^p$. The following proposition states the necessary optimality conditions for (14).

Proposition 1. *Let $\bar{t} < |\bar{x}|^p$, and (x^*, t^*) be an optimal solution of (14). Then, the following properties are satisfied*

(a) $\operatorname{sign}(x^*) = \operatorname{sign}(\bar{x})$,
(b) $t^* \geq \bar{t}$,
(c) $|x^*|^p \geq \bar{t}$,
(d) $t^* = |x^*|^p$.

Proof. We prove the statements by contradiction as follows:

(a) Suppose that $\operatorname{sign}(x^*) \neq \operatorname{sign}(\bar{x})$, then

$$|x^* - \bar{x}| = |x^* - 0| + |\bar{x} - 0| > |\bar{x} - 0|, \tag{15}$$

i.e., $(x^* - \bar{x})^2 > (0 - \bar{x})^2$. Hence, $g(x^*, t^*) - g(0, t^*) > 0$. Moreover, the feasibility of (x^*, t^*) implies that $t^* > 0$. Thus, $(0, t^*)$ is feasible and attains a lower objective value than that attained by (x^*, t^*). This contradicts the optimality of (x^*, t^*).

(b) Assume that $t^* < \bar{t}$. Then,

$$g(x^*, t^*) - g(x^*, \bar{t}) = (t^* - \bar{t})^2 > 0. \tag{16}$$

Furthermore, by the feasibility of (x^*, t^*), we have $|x^*|^p \leq t^* < \bar{t}$. Thus, $(x^*, \bar{t})$ is feasible and attains a lower objective value than that attained by (x^*, t^*). This contradicts the optimality of (x^*, t^*).

(c) Suppose that $|x^*|^p < \bar{t}$, i.e.,

$$-\bar{t}^{\frac{1}{p}} < x^* < \bar{t}^{\frac{1}{p}}. \tag{17}$$

We now consider two cases, $\bar{x} > 0$ and $\bar{x} < 0$. First, let $\bar{x} > 0$. Then, we have by (a) and (17) that $0 < x^* < \bar{t}^{\frac{1}{p}}$. Since $\bar{t} < |\bar{x}|^p$, i.e., $(\bar{x}, \bar{t}) \notin \mathcal{X}$, therefore $\bar{t}^{\frac{1}{p}} < \bar{x}$ and hence, $0 < x^* < \bar{t}^{\frac{1}{p}} < \bar{x}$. Pick $x_0 > 0$ such that $|x_0|^p = \bar{t}$, i.e., $x_0 = \bar{t}^{\frac{1}{p}}$. Then clearly, $x^* < x_0 < \bar{x}$. Thus, we have

$$g(x^*, t^*) - g(x_0, t^*) = (x^* - \bar{x})^2 - (x_0 - \bar{x})^2 > 0, \tag{18}$$

where the last inequality follows from the just proven identity that $x^* < x_0 < \bar{x}$. Moreover, we have by (b) that $|x_0|^p = \bar{t} \leq t^*$. Thus, (x_0, t^*) is feasible and attains a lower objective value than that attained by (x^*, t^*). This contradicts the optimality of (x^*, t^*).

On the other hand, let $\bar{x} < 0$. Then, we have by (a) and (17) that $-\bar{t}^{\frac{1}{p}} < x^* < 0$. Since $\bar{t} < |\bar{x}|^p$, i.e., $(\bar{x}, \bar{t}) \notin \mathcal{X}$, then $\bar{t}^{\frac{1}{p}} < |\bar{x}|$, i.e., $\bar{x} < -\bar{t}^{\frac{1}{p}}$. Therefore,

$$\bar{x} < -\bar{t}^{\frac{1}{p}} < x^*, \tag{19}$$

Pick $x_0 < 0$ such that $|x_0|^p = \bar{t}$, i.e., $x_0 = -\bar{t}^{\frac{1}{p}}$. Then, (18) also holds when $\bar{x} < 0$. Note that $|x_0|^p = \bar{t} \leq t^*$ by (b). Thus, (x_0, t^*) is feasible and attains a lower objective value than that attained by (x^*, t^*). This contradicts the optimality of (x^*, t^*).

(d) The feasibility of (x^*, t^*) eliminates the possibility that $t^* < |x^*|^p$. Now let $t^* > |x^*|^p$ and pick $t_0 = |x^*|^p$. Then, $\bar{t} \leq |x^*|^p = t_0 < t^*$, where the first inequality follows from (c). Then, $0 \leq t_0 - \bar{t} < t^* - \bar{t}$. Thus, we have

$$g(x^*, t^*) - g(x^*, t_0) = (t^* - \bar{t})^2 - (t_0 - \bar{t})^2 > 0, \tag{20}$$

Furthermore, the feasibility of (x^*, t_0) follows trivially from the choice of t_0. Thus, (x^*, t_0) is feasible and attains a lower objective value than that attained by (x^*, t^*). This contradicts the optimality of (x^*, t^*).

This concludes the proof. □

Algorithm 2: Nonconvex projection ($p = \frac{s}{q} < 1$)

$\mathcal{R} \leftarrow \operatorname{roots}\{a^{2q} + \frac{s}{q}(a^{2s} - \bar{t}a^s) - |\bar{x}|a^q\}$
$\bar{\mathcal{R}} \leftarrow \mathcal{R} \setminus \{\text{complex numbers and negative reals in } \mathcal{R}\}$
$\mathcal{T} \leftarrow \{(r^q, r^s) : r \in \bar{\mathcal{R}}\}$
$(\hat{x}, t^*) \leftarrow \operatorname{argmin}\{g(x,t) : (x,t) \in \mathcal{T}\}$
$x^* \leftarrow \operatorname{sign}(\bar{x})\hat{x}$

We now make use of the fact that for (14), an optimal solution (x^*, t^*) satisfies that $t^* = |x^*|^p$ and hence, (14) reduces to solving

$$\min_x \quad (|x|^p - \bar{t})^2 + (x - \bar{x})^2. \tag{21}$$

The first order necessary optimality condition for (21) implies the following:

$$p|x^*|^{p-1}\operatorname{sign}(x^*)(|x^*|^p - \bar{t}) + x^* - \bar{x} = 0. \tag{22}$$

By the symmetry of the function $|x|^p$, assume without loss of generality that $x^* > 0$ and let $0 < p = \frac{s}{q} < 1$ for some $s, q \in \mathbb{Z}_+$. A change of variables $a^q = x^*$ plugged in (22) shows that finding an optimal solution for (14) reduces to finding a root of the following scalar degree $2q$ polynomial:

$$a^{2q} + \frac{s}{q}\left(a^{2s} - \bar{t}a^s\right) - \bar{x}a^q. \tag{23}$$

Thus, to find $\Pi_{\mathcal{X}}(\bar{x}, \bar{t})$, solve for a root a^* of the polynomial in (23) such that (a^{*q}, a^{*s}) minimizes $g(x,t)$. Algorithm 2 summarizes the method we use to solve problem (14). In case $\bar{x} = 0$, we set $x^* = t^* = 0$. If the set $\bar{\mathcal{R}}$ is empty, we set $x^* = 0$ and $t^* = (\bar{t})^+$.

IV. Distributed convex projection

Step 4 of Algorithm 1 involves solving a convex projection problem. Although convex, solving this problem can potentially be a computational bottleneck in some applications, or not at all feasible in some others. For example, designing a classifier with a huge training set renders the projection problem computationally intensive. Moreover, if the data is distributed among multiple entities, i.e., each entity has a private local data set, then a centralized solution to the projection problem might not be a viable option.

In many applications, the function $f(\mathbf{x})$ in (1) evaluates the loss incurred by $\mathbf{x}$ averaged on a data set $\mathcal{T}$, i.e., f merits a separable structure with respect to the data. Thus, we propose to solve the projection problem using a scatter-gather type of

algorithm in which a master node distributes the computational load over a group of workers $\mathcal{N}$. Each worker performs a local *few-shot* learning, while the master combines their decisions.

Let $\mathcal{T} = \{(\mathbf{u}_j, v_j)\}_{j\in[m]}$ be a training set of feature-label pairs with m examples, and consider $f(\mathbf{x}) = \frac{1}{m}\sum_{j\in[m]} \ell(\mathbf{u}_j, v_j, \mathbf{x}) - \epsilon$, where ℓ is some convex loss function. Then, $\Pi_{\mathcal{Y}}(\bar{\mathbf{y}})$ entails solving:

$$\min_{\mathbf{y}} \quad \frac{1}{2}\|\mathbf{y}-\bar{\mathbf{y}}\|^2 \quad \text{s.t.} \quad \frac{1}{m}\sum_{j\in[m]} \ell(\mathbf{u}_j, v_j, \mathbf{y}) \leq \epsilon. \tag{24}$$

Define $\{\mathcal{T}_i\}_{i\in\mathcal{N}}$ as a partition of $\mathcal{T}$, where the ith set of training examples $\mathcal{T}_i$ is assigned to worker $i \in \mathcal{N}$. Indeed, $\mathcal{T}_i$ might be a local private data set generated at worker $i \in \mathcal{N}$. Instead of solving (24) in a centralized setting, the master broadcasts $\bar{\mathbf{y}}$ to all workers, each worker finds

$$\mathbf{y}_i^* = \underset{\mathbf{y}_i}{\operatorname{argmin}} \quad \frac{1}{2}\|\mathbf{y}_i-\bar{\mathbf{y}}\|^2 \quad \text{s.t.} \quad \frac{1}{|\mathcal{T}_i|}\sum_{j\in\mathcal{T}_i} \ell(\mathbf{u}_j, v_j, \mathbf{y}_i) \leq \epsilon, \tag{25}$$

reports its local decision $\mathbf{y}_i^*$ to the master, then the master combines the workers' decision simply via averaging, i.e., $\mathbf{y}^* = \sum_{i\in\mathcal{N}} \frac{|\mathcal{T}_i|}{m}\mathbf{y}_i^*$. This $\mathbf{y}^*$ is the updated $\mathbf{y}$ variable with which Algorithm 1 proceeds.

V. Numerical Results

We conduct numerical experiments on two families of problems: i) sparse signal reconstruction from noisy measurements, and ii) binary classification using support vector machines. We compare our method on problem (1) at $p = 0.5$ with a CVX [26] solution of (1) at $p = 1$.

A. Sparse signal reconstruction

Let $n = 2^{10}$, $m = \frac{n}{4}$, and randomly construct a matrix $\mathbf{U} \in \mathbb{R}^{m\times n}$ such that $\mathbf{U} = [\mathbf{M}, -\mathbf{M}]$ and $\mathbf{M}$ is a sparse binary random matrix with a few number of ones in each column. More precisely, the number of ones in each column of $\mathbf{M}$ is generated independently and randomly in the range of integers between 10 and 20, and their locations are randomly chosen independently for each column. Sparse binary random matrices satisfy a weak form of the RIP, and following the setup in [20], the concatenation of two copies of the same random matrix in $\mathbf{U}$ implies that column orthogonality is not maintained. A reference signal $\mathbf{x}_{\text{opt}}$ is generated such that $\|\mathbf{x}_{\text{opt}}\|_0 = \lceil 0.2n \rceil$. The indices of the nonzero entries of $\mathbf{x}_{\text{opt}}$ are uniform randomly chosen, and the value of each nonzero entry is generated according to a zero-mean unit-variance Gaussian distribution. The measurement vector $\mathbf{v} = \mathbf{U}\mathbf{x}_{\text{opt}} + \mathbf{n}$, where $\mathbf{n}$ is a zero-mean Gaussian noise vector with covariance $\sigma^2\mathbf{I}$, and $\mathbf{I}$ is the identity matrix. We reconstruct a sparse signal from the noisy measurements $\mathbf{v}$ via solving (1) with $f(\mathbf{x}) = \|\mathbf{U}\mathbf{x}-\mathbf{v}\|/\|\mathbf{v}\| - \epsilon$, i.e., $f(\mathbf{x})$ measures the relative residual corresponding to $\mathbf{x}$.

Fig. 1 plots the sparsity level of solutions obtained by both ℓ_p quasi-norm and ℓ_1 norm minimization at various noise variance levels. In particular, we solve (1) at $p = 1/2$ and $p = 1$. An entry of $\mathbf{x}$ whose absolute value exceeds 10^{-6} is considered a nonzero element and we compute the zero norm of $\mathbf{x}$ accordingly. At each value of σ^2, after the generation of the corresponding measurement vector $\mathbf{v}$, we choose $\epsilon = \|\mathbf{U}\mathbf{x}_{\text{opt}} - \mathbf{v}\|/\|\mathbf{v}\|$. Obviously, this choice of ϵ is not possible in practice as $\mathbf{x}_{\text{opt}}$ is not known a priori. Nevertheless, its choice is motivated by the inherent dependence of ϵ and σ^2. One chooses ϵ depending on the noise level, i.e., the higher the noise variance the higher the value of ϵ. Fig. 1 shows that our method produces sparser solutions than those obtained by ℓ_1 when the noise level is relatively high. It is worth mentioning that the solutions produced by Algorithm 1 are feasible with respect to (1). Furthermore, the behavior depicted by Fig. 1 persists for any problem instance generated as described above.

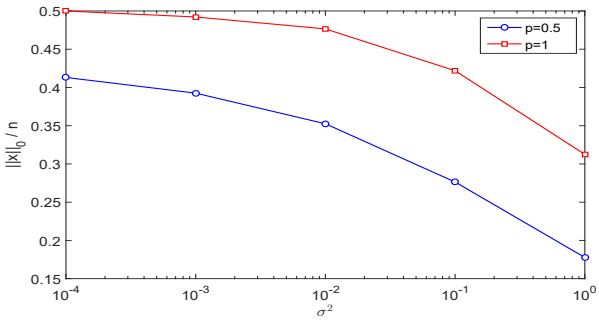

Fig. 1. Effect of noise variance on the sparsity of solutions obtained by ℓ_p quasi-norm and ℓ_1 norm minimization.

B. Binary classification

We use support vector machines to build a spam email classifier. The training set used is a subset of the SpamAssassin Public Corpus [24]. Let $\{(\mathbf{u}_j, v_j)\}_{j\in[m]}$ be the training set of feature vectors $\mathbf{u}_j \in \{0,1\}^n$ with corresponding labels $v_j \in \{-1, 1\}$ identifying whether the email is spam or not. We do not dwell on the details related to feature extraction since it is not the message that this experiment conveys. Instead, we highlight the effectiveness of our method in deciding whether an email is spam or not based on as few number of words as possible. Indeed, the body of an email is preprocessed based on ideas promoted by [25]. Following [25], we maintain a vocabulary list of $n = 1899$ words. Then, for a given preprocessed email $j \in [m]$, the wth entry of $\mathbf{u}_j$ is 1 if word w in the vocabulary list appears in email j, and it is zero otherwise. We build a linear classifier with a decision rule $\hat{v} = \text{sign}(\mathbf{u}^\top\mathbf{x})$, where $\mathbf{u}$ is the feature vector of the email in question, and $\mathbf{x}$ is the linear classifier's coefficients with its first entry being the bias term.

The purpose is to build a classifier that attains a high accuracy on the training set and simultaneously uses the least possible number of words from the vocabulary list to detect whether an email is spam or legit, i.e., a low-complexity classifier. We build the classifier by solving (1) with:

$$f(\mathbf{x}) = \frac{1}{m}\sum_{j\in[m]} \left(1 - v_j\mathbf{u}_j^\top\mathbf{x}\right)^+ - \epsilon. \tag{26}$$

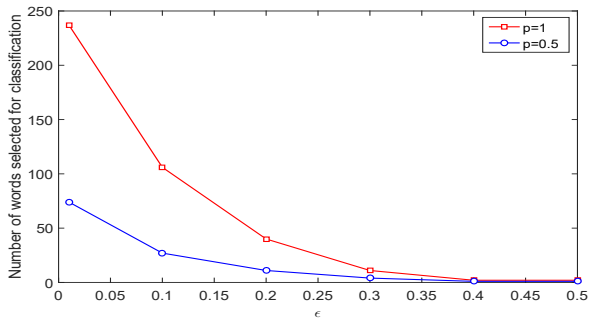

Fig. 2. Number of words selected for classification versus ϵ for ℓ_p quasi-norm and ℓ_1 norm minimization.

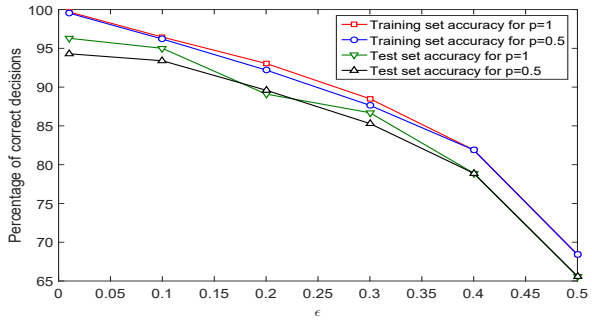

Fig. 3. Training and test set accuracies versus ϵ for ℓ_p quasi-norm and ℓ_1 norm minimization.

The desired training set accuracy is controlled by choosing ϵ, i.e., the lower the value of ϵ the higher the accuracy. Clearly, f not only urges $\mathbf{u}_j^\top \mathbf{x}$ to have the same sign as v_j, but it also urges that to happen with a margin, i.e., it tries to push $\mathbf{u}_j^\top \mathbf{x}$ below -1 if $v_j = -1$ to incur no cost from the jth training example. Likewise, it tries to push $\mathbf{u}_j^\top \mathbf{x}$ above 1 if $v_j = 1$.

We run Algorithm 1 with $p = 1/2$ on 2000 training emails at various values of ϵ. We terminate the algorithm after 100 iterations at each value of ϵ, and evaluate the performance of the learned classifier on a test set of 1000 emails. We solve (1) at $p = 1$ for comparing its performance with that obtained by Algorithm 1 at $p = 1/2$. Fig. 2 shows the number of nonzero entries in the classifier's coefficients $\mathbf{x}$ learned at $p = 1$ and $p = 1/2$. An entry of $\mathbf{x}$ is considered nonzero if its absolute value is above 10^{-4}. The corresponding training and test set accuracies for the obtained classifiers are plotted in Fig. 3. Fig. 2 and 3 together show that our method obtains classifiers that use significantly less number of words to make a decision on the legitimacy of an email while achieving almost similar level of accuracy on both the training and test sets. This behavior is consistent for all values of ϵ. For instance, at $\epsilon = 0.2$, $\ell_{0.5}$ minimization uses 11 words from the vocabulary list to make decisions as opposed to 40 words used by ℓ_1 minimization. Nevertheless, both $\ell_{0.5}$ and ℓ_1 attain about the same training and test set accuracies, 93% and 90%, respectively.

VI. Conclusion

We present a nonconvex ADMM algorithm that approximates the solution of the ℓ_p $(0 < p < 1)$ quasi-norm minimization problem. The algorithm is computationally efficient, where its complexity is bounded by the effort of finding a root for a scalar degree $2q$ polynomial for a rational $0 < p = s/q < 1$. The method is numerically shown to significantly outperform ℓ_1 in terms of sparsity of generated solutions at similar performance levels in different tasks.

References

[1] Cands E.J. and Tao T. Decoding by linear programming. IEEE transactions on information theory, 2005;51(12):4203-15.

[2] Chen S.S., Donoho D.L. and Saunders M.A. Atomic decomposition by basis pursuit. SIAM review, 2001;43(1):129-59.

[3] Taubman D.S. and Marcellin M.W. JPEG 2000: Image Compression Fundamentals. Standards and Practice, Kluwer Academic, Norwell, MA; 2001.

[4] Mallat S. A wavelet tour of signal processing. Academic press; 1999.

[5] Wright J., Yang A.Y., Ganesh A., Sastry S.S., Ma Y. Robust face recognition via sparse representation. IEEE transactions on pattern analysis and machine intelligence, 2009;31(2):210-27.

[6] Cands E.J., Romberg J. and Tao, T. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. IEEE Transactions on information theory, 2006;52(2):489-509.

[7] Donoho D.L. Compressed sensing. IEEE Transactions on information theory, 2006;52(4):1289-306.

[8] Bruckstein A.M., Donoho D.L. and Elad M. From sparse solutions of systems of equations to sparse modeling of signals and images. SIAM review, 2009;51(1):34-81.

[9] Natarajan B.K. Sparse approximate solutions to linear systems. SIAM journal on computing, 1995;24(2):227-34.

[10] Tropp J.A. and Wright S.J. Computational methods for sparse solution of linear inverse problems. Proceedings of the IEEE, 2010;98(6):948-58.

[11] Tropp J.A. Greed is good: Algorithmic results for sparse approximation. IEEE Transactions on Information theory, 2004;50(10):2231-42.

[12] Tibshirani R. Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society, Series B (Methodological), 1996;1:267-88.

[13] Chartrand R. Exact reconstruction of sparse signals via nonconvex minimization. IEEE Signal Processing Letters, 2007;14(10):707-10.

[14] Chartrand R. and Yin W. Iteratively reweighted algorithms for compressive sensing. IEEE international conference on acoustics, speech and signal processing, 2008, pp. 3869-3872.

[15] Bach F., Jenatton R., Mairal J. and Obozinski G. Optimization with sparsity-inducing penalties. Foundations and Trends in Machine Learning, 2012;4(1):1-106.

[16] Donoho D.L. For most large underdetermined systems of linear equations the minimal ℓ_1 norm solution is also the sparsest solution. Communications on pure and applied mathematics, 2006;59(6):797-829.

[17] Saab R, Chartrand R and Yilmaz O. Stable sparse approximations via nonconvex optimization. IEEE international conference on acoustics, speech and signal processing, 2008, pp. 3885-3888.

[18] Chartrand R. Fast algorithms for nonconvex compressive sensing: MRI reconstruction from very few data. IEEE international symposium on biomedical imaging, 2009, pp. 262-265.

[19] Mourad N.and Reilly J.P. Minimizing nonconvex functions for sparse vector reconstruction. IEEE Transactions on Signal Processing, 2010;58(7):3485-3496.

[20] Ge D., Jiang X. and Ye Y. A note on the complexity of Lp minimization. Mathematical programming, 2011;129(2):285-299.

[21] Chartrand R. and Wohlberg B. A nonconvex ADMM algorithm for group sparsity with sparse groups. In IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2013, pp. 6009-6013.

[22] Gorodnitsky I.F., and Rao B.D. Sparse signal reconstruction from limited data using FOCUSS: A re-weighted minimum norm algorithm. IEEE Transactions on Signal Processing, 1997;45(3):600-616.

[23] Boyd S., Parikh N., Chu E., Peleato B., Eckstein J. Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundations and Trends in Machine learning. 2011;3(1):1-22.

[24] SpamAssassin Public Corpus, http://spamassassin.apache.org

[25] Andrew Ng, Machine learning MOOC, available online: https://www.coursera.org/learn/machine-learning

[26] Michael Grant and Stephen Boyd. CVX: Matlab software for disciplined convex programming, version 2.0 beta. http://cvxr.com/cvx, 2013.