L_{1-2} Regularized Logistic Regression

Jing Qin*, and Yifei Lou†
*Department of Mathematics, University of Kentucky, Lexington, KY 40506, jing.qin@uky.edu
†Mathematical Sciences, The University of Texas at Dallas, Richardson, TX 75080, Yifei.Lou@utdallas.edu

***Abstract*—Logistic regression has become a fundamental tool to facilitate data analysis and prediction in a variety of applications, including health care and social sciences. Depending on different sparsity assumptions, logistic regression models often incorporate various regularizations, including ℓ_1-norm, ℓ_2-norm and some non-convex regularizations. In this paper, we propose a non-convex ℓ_{1-2}-regularized logistic regression model assuming that the coefficients to be recovered are highly sparse. We derive two numerical algorithms with guaranteed convergence based on the alternating direction method of multipliers and the proximal operator of ℓ_{1-2}. Numerical experiments on real data demonstrate the great potential of the proposed approach.**

I. Introduction

Logistic regression is one of the most fundamental statistical approaches for analyzing data and making practical predictions in physics, economy, medical science, social science, and other related fields. In principle, it estimates the probability of a response (an independent variable) based on the available features (dependent variables). In case of binomial logistic regression, the number of responses is restricted to two, e.g., hospitalized versus non-hospitalized in health prediction for patients.

Due to the limited number of training data, certain regularization technique is always necessary to avoid over-fitting in machine learning. More specifically, a generic regularized logistic regression seeks a hyperplane $\mathbf{w}^T\mathbf{x} + v = 0$ in $\mathbb{R}^n$ that separates the training data $\mathbf{x}_i \in \mathbb{R}^n$ for $i = 1, \ldots, m$ into two classes by solving the following minimization problem

$$\min_{\mathbf{w}\in\mathbb{R}^n, v\in\mathbb{R}} l_{avg}(\mathbf{w}, v) + \lambda J(\mathbf{w}), \tag{1}$$

where $J(\mathbf{w})$ is a regularization term, and $l_{avg}(\mathbf{w}, v)$ is the average loss function defined by [1]

$$l_{avg}(\mathbf{w}, v) = \frac{1}{m}\sum_{i=1}^{m}\left[y_i(\mathbf{x}_i^T\mathbf{w} + v) - \ln(1 + e^{\mathbf{w}^T\mathbf{x}_i + v})\right]. \tag{2}$$

Here $y_i \in \{1, -1\}$ is the class label associated with the i-th data instance $\mathbf{x}_i$. Based on the assumption of the data, e.g., sparsity and geometric characteristics, various regularizations as $J(\mathbf{w})$ have been applied to the model (1). By assuming that the underlying data follows the Gaussian distribution, the ℓ_2 regularization (a.k.a. ridge regularization) as a special case of Tikhonov regularization has been widely used in regression [2]. To utilize the sparse structure of data, least absolute shrinkage and selection operator (LASSO) used the ℓ_1 regularization in the least squares regression [3] which was later extended to many other related models, e.g., [4]. It has shown that the instance complexity grows in the number of irrelevant features logarithmically in the ℓ_1-regularized logistic regression but linearly in the ℓ_2-regularized one in [5]. Although it has many attractive properties, the ℓ_1 regularization results in a proximal operator (a.k.a. shrinkage) which causes significant bias toward zero for large regression coefficients. In order to reduce this bias, some works have been proposed, including the elastic net that takes a convex combination of ℓ_1 and ℓ_2 [6], the smoothly clipped absolute deviation (SCAD) penalty [7] and the minimax concave penalty (MCP) [8]. The non-convexity of penalty functions in SCAD and MCP brings computational difficulties to implement these regressions. By contrast, the convex elastic net regularized models can be efficiently solved by coordinate descent [1].

Recently, the ℓ_{1-2} regularization has been applied in image processing and compressive sensing, which yields favorable results [9]–[11]. The non-convex nature of the ℓ_{1-2} regularization results in numerical challenges and hence it is always desirable to design some efficient convergent algorithms to solve the related problems. For example, a fast implementation of minimizing ℓ_{1-2}-regularized model is proposed in [12]. All the aforementioned ℓ_{1-2}-regularized models [9]–[11] involve a quadratic data fitting term. In this paper, we incorporate the ℓ_{1-2}-regularization into a nonlinear logistic regression model

$$\min_{\mathbf{w}\in\mathbb{R}^n, v\in\mathbb{R}} l_{avg}(\mathbf{w}, v) + \lambda(\|\mathbf{w}\|_1 - \beta\|\mathbf{w}\|_2), \tag{3}$$

where $\beta \in [0, 1]$. Note that $J(\mathbf{w}) = \|\mathbf{w}\|_1 - \beta\|\mathbf{w}\|_2 \geq 0$ for any $\mathbf{w} \in \mathbb{R}^n$, but $J(\mathbf{w})$ is not a norm in $\mathbb{R}^n$. Furthermore, we propose two efficient algorithms based on the alternating direction method of multipliers (ADMM) [13], [14]. Refer to [15] for a more thorough discussion of ADMM. Each proposed algorithm consists of two subproblems: one subproblem is solved by Newton's method with line search and the other subproblem is solved by the proximal operator of ℓ_{1-2} in closed form. The difference of them lies in solving the $(\mathbf{w}, v)$-subproblem. In particular, we apply the alternating minimization so that $\mathbf{w}$ and v are solved separately by Newton's method with line search. To further subproblem errors and improve performance, we reformulate the model such that $(\mathbf{w}, v)$ can be treated as one unknown variable. Numerical experiments demonstrate that both algorithms can achieve high prediction accuracy and the second one performs better than the first one in terms of accuracy and efficiency.

The organization of the paper is as follows. Section II presents an efficient algorithm for solving ℓ_{1-2}-regularized logistic regression. Section III demonstrates the performance of

the proposed method in data classification. Brief conclusions and future work are provided in Section IV.

II. Proposed Algorithms

A. Proximal Operator of ℓ_{1-2}

To make the paper self-contained, we derive the closed-form expression for the proximal operator of ℓ_{1-2}, based on that of the ℓ_1-norm. Note that this proof is different from that in [11].

Lemma 1. *The proximal operator of the ℓ_1-norm, i.e., the solution to*

$$\min_{\mathbf{x}\in\mathbb{R}^n} \lambda \|\mathbf{x}\|_1 + \frac{1}{2}\|\mathbf{x}-\mathbf{b}\|_2^2 = \sum_{i=1}^{n}\left(\lambda|x_i| + \frac{1}{2}(x_i-b_i)^2\right) \tag{4}$$

with $\lambda > 0$ is given by

$$\mathrm{prox}_{\lambda\|\cdot\|_1}(\mathbf{b}) = \mathrm{sign}(\mathbf{b}).*\max\{|\mathbf{b}|-\lambda, 0\}. \tag{5}$$

Proof. Since the objective function in (4) is componentwise separable, we can get the optimality condition for a specific component x_i, that is,

$$\lambda\,\mathrm{sign}(x_i) + x_i - b_i = 0.$$

Due to the fact that $x_i = \mathrm{sign}(x_i)|x_i|$, we have

$$\mathrm{sign}(x_i)(\lambda + |x_i|) = b_i.$$

Since $\lambda > 0$ and $|x_i| \geq 0$, we have $\mathrm{sign}(x_i) = \mathrm{sign}(b_i)$ and $\lambda + |x_i| = |b_i|$. If $|b_i| \geq \lambda$, then we get

$$x_i = \mathrm{sign}(x_i)|x_i| = \mathrm{sign}(b_i)(|b_i|-\lambda).$$

Otherwise if $\lambda > |b_i|$, we have $x_i = \mathbf{0}$. In summary, we have

$$\mathbf{x} = \mathrm{sign}(\mathbf{b}).*\max\{|\mathbf{b}|-\lambda, 0\},$$

where $.*$ is the componentwise multiplication. □

Theorem 1. *The proximal operator for the ℓ_{1-2} semi-norm, i.e., the solution to the problem*

$$\min_{\mathbf{x}\in\mathbb{R}^n} \lambda(\|\mathbf{x}\|_1 - \beta\|\mathbf{x}\|_2) + \frac{1}{2}\|\mathbf{x}-\mathbf{b}\|_2^2,$$

with $\lambda > 0$ and $\beta \in (0, 1]$ has the form

$$\mathrm{prox}_{\lambda(\|\cdot\|_1-\beta\|\cdot\|_2)}(\mathbf{b}) = \mathbf{z} + \lambda\beta\frac{\mathbf{z}}{\|\mathbf{z}\|_2}, \tag{6}$$

where $\mathbf{z} = \mathrm{prox}_{\lambda\|\cdot\|_1}(\mathbf{b})$.

Proof. Consider the i-th optimality condition when $\mathbf{x} \neq \mathbf{0}$

$$\mathrm{sign}(x_i)\left(\lambda - \lambda\beta\frac{|x_i|}{\|\mathbf{x}\|_2} + |x_i|\right) = b_i. \tag{7}$$

Next we construct x_i based on the proximal operator of ℓ_1. Let $\mathbf{z} = \mathrm{prox}_{\lambda\|\cdot\|_1}(\mathbf{b})$ which implies

$$\lambda\,\mathrm{sign}(z_i) + z_i - b_i = 0, \quad i = 1, \ldots, n.$$

Based on the proof of Lemma 1, we get $\mathrm{sign}(z_i) = \mathrm{sign}(b_i)$ and $\lambda + |z_i| = |b_i|$ for $i = 1, \ldots, n$. Therefore, we have

$$\lambda - \lambda\beta\frac{|z_i|}{\|\mathbf{z}\|_2} + |z_i|(1 + \frac{\lambda\beta}{\|\mathbf{z}\|_2}) = \lambda + |z_i| = |b_i|.$$

Now we define

$$x_i = z_i + \lambda\beta z_i/\|\mathbf{z}\|_2 = z_i(1 + \lambda\beta/\|\mathbf{z}\|_2),$$

which satisfies that $\mathrm{sign}(x_i) = \mathrm{sign}(z_i)$ and

$$|x_i| = |z_i|(1 + \lambda\beta/\|\mathbf{z}\|_2).$$

Since $\|\mathbf{x}\|_2^2 = \sum_{i=1}^n |x_i|^2$, we get $\|\mathbf{x}\|_2 = \|\mathbf{z}\|_2 + \lambda\beta$. Then the division of $|x_i|$ by $\|\mathbf{x}\|_2$ results in the relation

$$\frac{|x_i|}{\|\mathbf{x}\|_2} = \frac{|z_i|}{\|\mathbf{z}\|_2}.$$

Thus

$$\lambda - \lambda\beta\frac{|x_i|}{\|\mathbf{x}\|_2} + |x_i| = \lambda - \lambda\beta\frac{|z_i|}{\|\mathbf{z}\|_2} + |z_i|(1 + \frac{\lambda\beta}{\|\mathbf{z}\|_2}) = |b_i|.$$

Combination of the above equation and $\mathrm{sign}(x_i) = \mathrm{sign}(z_i) = \mathrm{sign}(b_i)$ yields (7), which completes the proof. □

B. First Proposed Algorithm

We first introduce a new variable $\mathbf{z} \in \mathbb{R}^n$ and rewrite (3) as follows

$$\min_{\mathbf{w}\in\mathbb{R}^n, v\in\mathbb{R}} l_{avg}(\mathbf{w}, v) + \lambda(\|\mathbf{z}\|_1 - \beta\|\mathbf{z}\|_2) \quad \text{s.t.} \quad \mathbf{z} = \mathbf{w}. \tag{8}$$

Then we define the augmented Lagrangian function

$$\begin{aligned}\mathcal{L}(\mathbf{w}, v, \mathbf{u}, \mathbf{z}) =& l_{avg}(\mathbf{w}, v) + \lambda(\|\mathbf{z}\|_1 - \beta\|\mathbf{z}\|_2) \\ &+ \frac{\rho}{2}\|\mathbf{z} - \mathbf{w} + \mathbf{u}\|_2^2.\end{aligned}$$

After applying ADMM, we get the following algorithm

$$\begin{cases} (\mathbf{w}^{k+1}, v^{k+1}) = \underset{\mathbf{w}\in\mathbb{R}^n, v}{\mathrm{argmin}} f(\mathbf{w}, v), \\ \mathbf{z}^{k+1} \in \underset{\mathbf{z}\in\mathbb{R}^n}{\mathrm{argmin}} \lambda(\|\mathbf{z}\|_1 - \beta\|\mathbf{z}\|_2) + \frac{\rho}{2}\|\mathbf{z} - \mathbf{w}^{k+1} + \mathbf{u}^k\|_2^2, \\ \mathbf{u}^{k+1} = \mathbf{u}^k + \gamma(\mathbf{z}^{k+1} - \mathbf{w}^{k+1}). \end{cases}$$

where

$$f(\mathbf{w}, v) = l_{avg}(\mathbf{w}, v) + \frac{\rho}{2}\|\mathbf{z}^k - \mathbf{w} + \mathbf{u}^k\|_2^2.$$

In the first $(\mathbf{w}, v)$-subproblem, the objective function is continuously differentiable and convex, which guarantees that Newton's method can provide a convergent result. Since the translation coefficient $v \in \mathbb{R}$ only appears in the function l_{avg}, the $(\mathbf{w}, v)$-subproblem can be further separated into the $\mathbf{w}$-subproblem and the v-subproblem by alternating minimization. By direct computation, we get the first and second derivatives of l_{avg} with respect to components of $\mathbf{w}$ as follows

$$\begin{aligned}\frac{\partial l_{avg}}{\partial w_j} &= \frac{1}{m}\sum_{i=1}^{m}\left[y_i X_{ij} - \frac{X_{ij}}{1 + e^{-\mathbf{x}_i^T\mathbf{w} - v}}\right], \\ \frac{\partial^2 l_{avg}}{\partial w_j \partial w_k} &= \frac{1}{m}\sum_{i=1}^{m}\frac{X_{ij}X_{ik}}{2 + e^{\mathbf{x}_i^T\mathbf{w}+v} + e^{-\mathbf{x}_i^T\mathbf{w}-v}},\end{aligned} \tag{9}$$

for $j, k = 1, \ldots, n$. Here $X \in \mathbb{R}^{m\times n}$ with X_{ij} as the j-th component of $\mathbf{x}_i$. Given $v, \mathbf{z}, \mathbf{u}$, the Newton's method with line search updates $\mathbf{w}$ at the t-th inner iteration is given by

$$\begin{aligned}\delta_{\mathbf{w}} &= -(\nabla^2_{\mathbf{w}} l_{avg}(\mathbf{w}^t, v) + \rho I_m)^{-1}\big(\nabla_{\mathbf{w}} l_{avg}(\mathbf{w}^t, v) \\ &\quad + \rho(\mathbf{w}^t - \mathbf{z} - \mathbf{u})\big) \\ \mathbf{w}^{t+1} &= \mathbf{w}^t + s_t\delta_{\mathbf{w}},\end{aligned}$$

where the step size s_t is the smallest value of the form θ^C with $\theta \in (0,1)$ and $C \in \mathbb{N}$ such that the objective function value at $\mathbf{w}^{t+1}$ is smaller than that at $\mathbf{w}^t$. Likewise, we apply the Newton's method with line search to solve the one-dimensional v-subproblem. Based on Theorem 1, the $\mathbf{z}$-subproblem can be explicitly solved by

$$\mathbf{z}^{k+1} = \operatorname{prox}_{\frac{\lambda}{\rho}(\|\cdot\|_1 - \beta\|\cdot\|_2)}(\mathbf{w}^{k+1} - \mathbf{u}^k). \tag{10}$$

We detail the corresponding algorithm in Algorithm 1.

Algorithm 1 L_{1-2}-regularized Logistic Regression

Require: data $X \in \mathbb{R}^{m\times n}$ and $\mathbf{y} \in \mathbb{R}^m$, parameters $\lambda, \rho > 0$, $\beta \in (0,1]$, $\gamma \in (0, \frac{\sqrt{5}+1}{2}]$, $\alpha > 0$ and $\theta \in (0,1)$, set the maximal number of inner and outer iterations N_{in}, N_{out}, and the tolerance $\varepsilon > 0$ for the stopping criteria.
Initialize: set $\mathbf{w}^0 = \mathbf{0}$, $v^0 = 0$.
for $k = 0, 1, \ldots, N_{out} - 1$ **do**
  $\widehat{\mathbf{w}}^0 = \mathbf{0}$, $\hat{v}^0 = 0$
  for $t = 1, \ldots, N_{in}$ **do** (Solve the $\mathbf{w}$-subproblem)
    $\mathbf{g} = \nabla_{\mathbf{w}} l_{avg}(\widehat{\mathbf{w}}^t, v^k) + \rho(\widehat{\mathbf{w}}^t - \mathbf{z}^k - \mathbf{u}^k)$
    $H = \nabla^2_{\mathbf{ww}} l_{avg}(\widehat{\mathbf{w}}^t, v^k) + \rho I_m$
    $\delta_{\mathbf{w}} = -H^{-1}\mathbf{g}$, $\delta_f = \mathbf{g}^T\delta_{\mathbf{w}}$, and $s_t = 1$
    while $f(\widehat{\mathbf{w}}^t + s_t\delta_{\mathbf{w}}, v^k) > f(\widehat{\mathbf{w}}^t, v^k) + \alpha s_t\delta_f$ **do**
      $s_t \leftarrow \theta s_t$
    end while
    $\widehat{\mathbf{w}}^{t+1} = \widehat{\mathbf{w}}^t + s_t\delta_{\mathbf{w}}$
  end for
  $\mathbf{w}^{k+1} = \widehat{\mathbf{w}}^{N_{in}}$
  for $t = 1, \ldots, N_{in}$ **do** (Solve the v-subproblem)
    $g = \nabla_v l_{avg}(\mathbf{w}^{k+1}, v^t)$
    $h = \nabla^2_{vv} l_{avg}(\mathbf{w}^{k+1}, v^t)$
    $\delta_v = -h^{-1}g$, $\delta_f = g\delta_v$, and $s_t = 1$
    while $l_{avg}(\mathbf{w}^{k+1}, \hat{v} + s_t\delta_v) > l_{avg}(\mathbf{w}^{k+1}, \hat{v}) + \alpha s_t\delta_f$ **do**
      $s_t \leftarrow \theta s_t$
    end while
    $\hat{v}^{t+1} = \hat{v}^t + s_t\delta_v$
  end for
  $v^{k+1} = \hat{v}^{N_{in}}$
  $\mathbf{z}^{k+1}$ is given by (10).
  $\mathbf{u}^{k+1} = \mathbf{u}^k + \gamma(\mathbf{z}^{k+1} - \mathbf{w}^{k+1})$
  If $(\frac{\|\mathbf{w}^{k+1} - \mathbf{w}^k\|_2^2 + (v^{k+1} - v^k)^2}{\|\mathbf{w}^k\|_2^2 + (v^k)^2})^{1/2} < \varepsilon$, then exit.
end for

C. Second Proposed Algorithm

We can see that Algorithm 1 involves two inner loops based on Newton's method. To further improve the computational efficiency and reduce errors in subproblems, we intend to combine the $\mathbf{w}$-subproblem and the v-subproblem. We first let $\mathbf{r} = (\mathbf{w}^T, v)^T \in \mathbb{R}^{n+1}$ and add a column of zeros to the end of the original data matrix $X \in \mathbb{R}^{m\times n}$. The resultant matrix is denoted by $\widetilde{X} \in \mathbb{R}^{m\times(n+1)}$. Then the calculations in (9) are still valid for $i, j = 1, \ldots, n+1$. Now we rewrite (3) as

$$\min_{\mathbf{r}\in\mathbb{R}^{n+1}} l_{avg}(\mathbf{r}) + \lambda(\|\mathbf{z}_{1:n}\|_1 - \beta\|\mathbf{z}_{1:n}\|_2), \quad \text{s.t.} \quad \mathbf{z} = \mathbf{r}, \tag{11}$$

where $\mathbf{z}_{1:n}$ is a sub-vector of $\mathbf{z}$ that consists of the first n components of $\mathbf{z}$. In this new formulation, $\mathbf{w}$ and v can be updated simultaneously with higher accuracy in the subproblems. We then apply ADMM to get the following algorithm

$$\begin{cases} \mathbf{r}^{k+1} = \operatorname*{argmin}_{\mathbf{r}\in\mathbb{R}^{n+1}} l_{avg}(\mathbf{r}) + \dfrac{\rho}{2}\left\|\mathbf{z}^k - \mathbf{r} + \mathbf{u}^k\right\|_2^2, \\ \mathbf{z}^{k+1} \in \operatorname*{argmin}_{\mathbf{z}\in\mathbb{R}^{n+1}} \lambda(\|\mathbf{z}_{1:n}\|_1 - \beta\|\mathbf{z}_{1:n}\|_2) \\ \qquad\qquad + \frac{\rho}{2}\left\|\mathbf{z} - \mathbf{r}^{k+1} + \mathbf{u}^k\right\|_2^2, \\ \mathbf{u}^{k+1} = \mathbf{u}^k + \gamma(\mathbf{z}^{k+1} - \mathbf{r}^{k+1}). \end{cases}$$

Note that the $\mathbf{z}$-subproblem can be further split into the $\mathbf{z}_{1:n}$-subproblem and z_{n+1}-subproblem where z_{n+1} is the $(n+1)$-st component of $\mathbf{z}$. Here $l_{avg}(\mathbf{r}) = l_{avg}(\mathbf{w}, v)$ and the $\mathbf{r}$-subproblem can be solved by Newton's method similar to Algorithm 1. We denote

$$\begin{aligned} F(\mathbf{r}) = \frac{1}{m}\sum_{i=1}^m \left[y_i(\widetilde{\mathbf{x}}_i^T\mathbf{r}) - \ln(1 + e^{\widetilde{\mathbf{x}}_i^T\mathbf{r}})\right] \\ + \frac{\rho}{2}\left\|\mathbf{z} - \bar{\mathbf{r}}^{k+1} + \mathbf{u}^k\right\|_2^2. \end{aligned}$$

The corresponding algorithm is detailed in Algorithm 2. Convergence analysis of both algorithms can be derived based on [12] for the proximal operator of ℓ_{1-2} and [15] for the general ADMM framework.

III. Numerical Results

In this section, we compare our proposed algorithms with other related methods in terms of regularization paths and accuracy. The three sets of test data are downloaded from the UCI machine learning repository[1]: Hepatitis, ionosphere, and spambase. We also test the microarray data[2] by Alon et al. [16]. The number of features and the number of instances for all data sets are listed in Table I. Following the standard data pre-processing, we remove the instances which contain missing features, and normalize each data set column-wise to have unit norm. To compare the prediction accuracy for each method, we use the area under the receiver operating characteristic (ROC) curve, denoted as AUC, which is estimated by `perfcurve` in Matlab. The AUC values vary in $[0, 1]$ and larger AUC value corresponds to higher prediction accuracy. All experiments were run in MATLAB 2016b on a desktop

[1] http://archive.ics.uci.edu/ml/datasets.html
[2] https://github.com/ramhiser/datamicroarray/tree/master/data

Algorithm 2 Improved L_{1-2}-regularized Logistic Regression

Require: data $X \in \mathbb{R}^{m\times n}$ and $\mathbf{y} \in \mathbb{R}^m$, parameters $\lambda, \rho > 0$, $\beta \in (0,1]$, $\gamma \in (0, \frac{\sqrt{5}+1}{2}]$, $\alpha > 0$ and $\theta \in (0,1)$, set the maximal number of inner and outer iterations N_{in}, N_{out}, and the tolerance $\varepsilon > 0$ for the stopping criteria.

Initialize: set $\mathbf{r}^0 = \mathbf{0}$.

for $k = 0, 1, \ldots, N_{out} - 1$ **do**

  $\widehat{\mathbf{r}}^0 = \mathbf{0}$

  for $t = 1, \ldots, N_{in}$ **do** (Solve the $\mathbf{r}$-subproblem)

    $\mathbf{g} = \nabla_{\mathbf{r}} F(\widehat{\mathbf{r}}^t) + \rho(\widehat{\mathbf{w}}^t - \bar{\mathbf{z}}^k - \mathbf{u}^k)$

    $H = \nabla^2_{\mathbf{rr}} F(\widehat{\mathbf{r}}^t) + \rho I_m$

    $\delta_{\mathbf{r}} = -H^{-1}\mathbf{g}$, $\delta_F = \mathbf{g}^T \delta_{\mathbf{r}}$, and $s_t = 1$

    while $F(\widehat{\mathbf{r}}^t + s_t\delta_{\mathbf{r}}) > F(\widehat{\mathbf{r}}^t) + \alpha s_t \delta_F$ **do**

      $s_t \leftarrow \theta s_t$

    end while

    $\widehat{\mathbf{r}}^{t+1} = \widehat{\mathbf{r}}^t + s_t\delta_{\mathbf{r}}$

  end for

  $\mathbf{r}^{k+1} = \widehat{\mathbf{r}}^{N_{in}}$

  $\mathbf{z}^{k+1}_{1:n} = \mathrm{prox}_{\frac{\lambda}{\rho}(\|\cdot\|_1 - \beta\|\cdot\|_2)}(\mathbf{r}^{k+1}_{1:n} - \mathbf{u}^k_{1:n})$

  $z^{k+1}_{n+1} = r^{k+1}_{n+1} - u^k_{n+1}$

  $\mathbf{u}^{k+1} = \mathbf{u}^k + \gamma(\mathbf{z}^{k+1} - \mathbf{r}^{k+1})$

  If $\frac{\|\mathbf{r}^{k+1} - \mathbf{r}^k\|_2}{\|\mathbf{r}^k\|_2} < \varepsilon$, then exit.

end for

computer with 64GB RAM and a 2.2GHz Intel Xeon CPU E5-2650 v4.

Data Name	No. of Features	No. of instances
Hepatitis	19	80
Ionosphere	32	351
Microarray	2000	62
Spambase	57	4601

TABLE I
DATASETS USED IN OUR EXPERIMENTS.

We compare our results with those given by the Matlab built-in function `lassoglm` and one widely used Matlab toolbox `glmnet` [17]. In particular, the former solves the LASSO ℓ_1-regularized logistic regression model using the iteratively reweighted least squares (IRLS) method [18], [19]. The latter solves the general elastic net regularized logistic regression model by coordinate descent [1]. Notice that `glmnet` has been highly optimized with major computation subroutine written in FORTRAN and compiled as a MEX file in Matlab. In our proposed algorithms, we set the maximum outer iteration number as 100, the maximum inner iteration number as 50, $\rho = 10^{-6}$, $\gamma = 1$ and $\varepsilon = 10^{-4}$.

In our first experiment, we test `glmnet` and Algorithm 2 on the ionosphere data, where the original first two features are removed as suggested by Matlab `lassoglm`. We choose 19 values for the regularization parameter λ in (1) evenly distributed between $10^{-4.2}$ to $10^{-0.6}$ in the logarithmic scale. In Figure 1, we display the regularization paths for different regularizations, where each figure contains 32 coefficient trajectories (excluding the translation coefficient v) with respect to λ. In the extreme case when $\beta = 0$, the ℓ_{1-2}-regularization reduces to the ℓ_1-regularization, i.e., LASSO. One can see that all trajectories become flat as the value of λ decreases but may oscillate for relatively large values of λ. The elastic net regularization, essentially the convex combination of ℓ_1-norm and ℓ_2-norm squared, enforces smoothness for all coefficient trajectories. In the meanwhile, the ℓ_{1-2}-regularization allows more non-smooth coefficient trajectories or fluctuations on some interval of λ when β approaches one. This implies that the ℓ_{1-2}-regularization can handle more inhomogeneous cases. Moreover, the elastic net profile shows the grouping effect, i.e., strongly correlated coefficients tend to be in or out together [6], which is not obvious in the ℓ_{1-2} profile.

In our second experiment, we choose 25 values logarithmically spaced in the interval $(10^{-4}, 1)$ as the regularization parameter λ. For each algorithm, we run the k-fold cross-validation with $k = 10$ to find the optimal λ. Table II compares the AUC values for all competing methods on various data sets. From the results, we can see that our proposed Algorithm 2 has the highest accuracy. Algorithm 2 performs better than Algorithm 1 in terms of prediction accuracy and computational efficiency. In addition, if the number of features is significantly larger than that of instances, e.g., the microarray data, most methods can get results with high accuracy. If the number of instances is insufficient, e.g., the hepatitis data, it becomes more difficult to get accurate prediction. Not only does Algorithm 2 performs better than Algorithm 1 in terms of prediction accuracy, it is also faster. However, both proposed algorithms are slower than `glmnet`, due to high computational cost in calculating the gradient, the Hessian matrix and its pseudo-inverse. It is our future work to design a more efficient algorithm for solving ℓ_{1-2}-regularized problems.

Method	Hepatitis	Ionosphere	Microarray	Spambase
LASSO	0.8530	0.9398	0.9830	0.9761
glmnet	0.8859	0.9661	1.0000	0.9765
Alg.1	0.8852	0.9661	1.0000	0.9764
Alg.2	0.8859	0.9661	1.0000	0.9774

TABLE II
COMPARISON OF AUC VALUES FOR VARIOUS METHODS AND DATASETS.

IV. CONCLUSIONS

In this work, we propose a ℓ_{1-2}-regularized logistic regression model where the regularization term is a difference of ℓ_1-norm and ℓ_2-norm. Although this model is non-convex which causes computation and implementation challenges, we propose two numerical algorithms based on the framework of ADMM and the proximal operator of ℓ_{1-2}. Specifically, we first derive a numerical algorithm that involves alternating minimization, Newton's method and the ℓ_{1-2} proximal operator. To further improve the computational efficiency, we combine all coefficients as one variable, and derive another algorithm by reformulating the minimization problem. Numerical experiments have shown that the proposed algorithms can achieve the state-of-the-art accuracy in terms of AUC. In the future

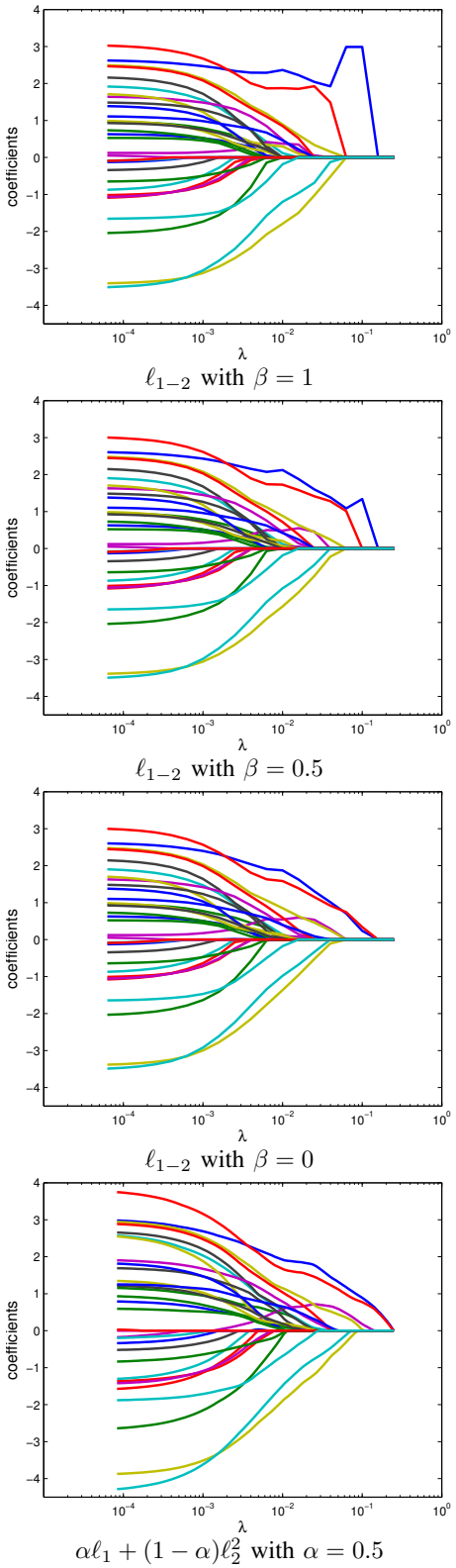

Fig. 1. Regularization path for the ionosphere data. From top to bottom: ℓ_{1-2} regularization with $\beta = 1, 0.5, 0$ and elastic net regularization with $\alpha = 0.5$ by `glmnet`.

work, we will further enhance the computational efficiency of the proposed algorithms in coordinate updating manner and extend them to solve other related regression problems.

V. Acknowledgements

J. Qin is supported by the NSF grant DMS-1941197, and Y. Lou is supported by the NSF grants DMS-1522786, CAREER DMS-1846690.

References

[1] J. Friedman, T. Hastie, and R. Tibshirani, "Regularization paths for generalized linear models via coordinate descent," *Journal of statistical software*, vol. 33, no. 1, p. 1, 2010.

[2] K. Nigam, J. Lafferty, and A. McCallum, "Using maximum entropy for text classification," in *IJCAI-99 workshop on machine learning for information filtering*, vol. 1, 1999, pp. 61–67.

[3] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

[4] ——, "The lasso method for variable selection in the cox model," *Statistics in medicine*, vol. 16, no. 4, pp. 385–395, 1997.

[5] A. Y. Ng, "Feature selection, L_1 vs. L_2 regularization, and rotational invariance," in *Proceedings of the twenty-first international conference on Machine learning*. ACM, 2004, p. 78.

[6] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 67, no. 2, pp. 301–320, 2005.

[7] J. Fan and R. Li, "Variable selection via nonconcave penalized likelihood and its oracle properties," *Journal of the American statistical Association*, vol. 96, no. 456, pp. 1348–1360, 2001.

[8] C.-H. Zhang, "Nearly unbiased variable selection under minimax concave penalty," *The Annals of statistics*, pp. 894–942, 2010.

[9] Y. Lou, T. Zeng, S. Osher, and J. Xin, "A weighted difference of anisotropic and isotropic total variation model for image processing," *SIAM Journal on Imaging Sciences*, vol. 8, no. 3, pp. 1798–1823, 2015.

[10] P. Yin, Y. Lou, Q. He, and J. Xin, "Minimization of ℓ_{1-2} for compressed sensing," *SIAM Journal on Scientific Computing*, vol. 37, no. 1, pp. A536–A563, 2015.

[11] Y. Lou, P. Yin, Q. He, and J. Xin, "Computing sparse representation in a highly coherent dictionary based on difference of l_1 and l_2," *Journal of Scientific Computing*, vol. 64, no. 1, pp. 178–196, 2015.

[12] Y. Lou and M. Yan, "Fast L1–L2 minimization via a proximal operator," *Journal of Scientific Computing*, vol. 74, no. 2, pp. 767–785, 2018.

[13] R. Glowinski and A. Marroco, "Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de dirichlet non linéaires," *Revue française d'automatique, informatique, recherche opérationnelle. Analyse numérique*, vol. 9, no. R2, pp. 41–76, 1975.

[14] D. Gabay and B. Mercier, "A dual algorithm for the solution of nonlinear variational problems via finite element approximation," *Computers & Mathematics with Applications*, vol. 2, no. 1, pp. 17–40, 1976.

[15] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[16] U. Alon, N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack, and A. J. Levine, "Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays," *Proceedings of the National Academy of Sciences*, vol. 96, no. 12, pp. 6745–6750, 1999.

[17] J. Qian, T. Hastie, J. Friedman, R. Tibshirani, and N. Simon, "Glmnet for Matlab," *http://www.stanford.edu/~hastie/glmnet_matlab*, 2013.

[18] P. J. Green, "Iteratively reweighted least squares for maximum likelihood estimation, and some robust and resistant alternatives," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 149–192, 1984.

[19] T. P. Minka, "A comparison of numerical optimizers for logistic regression," *https://tminka.github.io/papers/logreg/*, 2003.