Journal of Machine Learning Research 12 (2011) 1537-1586 Submitted 5/10; Revised 12/10; Published 5/11

# Super-Linear Convergence of Dual Augmented Lagrangian Algorithm for Sparsity Regularized Estimation

**Ryota Tomioka** TOMIOKA@MIST.I.U-TOKYO.AC.JP
**Taiji Suzuki** S-TAIJI@STAT.T.U-TOKYO.AC.JP
*Department of Mathematical Informatics*
*The University of Tokyo*
*7-3-1, Hongo, Bunkyo-ku, Tokyo, 113-8656, Japan*

**Masashi Sugiyama** SUGI@CS.TITECH.AC.JP
*Department of Computer Science*
*Tokyo Institute of Technology*
*2-12-1-W8-74, O-okayama, Meguro-ku, Tokyo, 152-8552, Japan*

**Editor:** Tong Zhang

## Abstract

We analyze the convergence behaviour of a recently proposed algorithm for regularized estimation called Dual Augmented Lagrangian (DAL). Our analysis is based on a new interpretation of DAL as a proximal minimization algorithm. We theoretically show under some conditions that DAL converges super-linearly in a non-asymptotic and global sense. Due to a special modelling of sparse estimation problems in the context of machine learning, the assumptions we make are milder and more natural than those made in conventional analysis of augmented Lagrangian algorithms. In addition, the new interpretation enables us to generalize DAL to wide varieties of sparse estimation problems. We experimentally confirm our analysis in a large scale $\ell_1$-regularized logistic regression problem and extensively compare the efficiency of DAL algorithm to previously proposed algorithms on both synthetic and benchmark data sets.

**Keywords:** dual augmented Lagrangian, proximal minimization, global convergence, sparse estimation, convex optimization

## 1. Introduction

Sparse estimation through convex regularization has become a common practice in many application areas including bioinformatics and natural language processing. However facing the rapid increase in the size of data-sets that we analyze everyday, clearly needed is the development of optimization algorithms that are tailored for machine learning applications.

Regularization-based sparse estimation methods estimate unknown variables through the minimization of a loss term (or a data-fit term) plus a regularization term. In this paper, we focus on convex methods; that is, both the loss term and the regularization term are convex functions of unknown variables. Regularizers may be nondifferentiable on some points; the nondifferentiability can promote various types of sparsity on the solution.

Although the problem is convex, there are three factors that challenge the straight-forward application of general tools for convex optimization (Boyd and Vandenberghe, 2004) in the context of machine learning.

The first factor is the diversity of loss functions. Arguably the squared loss is most commonly used in the field of signal/image reconstruction, in which many algorithms for sparse estimation have been developed (Figueiredo and Nowak, 2003; Daubechies et al., 2004; Cai et al., 2008). However the variety of loss functions is much wider in machine learning, to name a few, logistic loss and other log-linear loss functions. Note that these functions are not necessarily strongly convex like the squared loss. See Table 1 for a list of loss functions that we consider.

The second factor is the nature of the data matrix, which we call the design matrix in this paper. For a regression problem, the design matrix is defined by stacking input vectors along rows. If the input vectors are numerical (e.g., gene expression data), the design matrix is dense and has no structure. In addition, the characteristics of the matrix (e.g., the condition number) is unknown until the data is provided. Therefore, we would like to minimize assumptions about the design matrix, such as, sparse, structured, or well conditioned.

The third factor is the large number of unknown variables (or parameters) compared to observations. This is a situation regularized estimation methods are commonly applied. This factor may have been overlooked in the context of signal denoising, in which the number of observations and the number of parameters are equal.

Various methods have been proposed for efficient sparse estimation (see Figueiredo and Nowak, 2003; Daubechies et al., 2004; Combettes and Wajs, 2005; Andrew and Gao, 2007; Koh et al., 2007; Wright et al., 2009; Beck and Teboulle, 2009; Yu et al., 2010, and the references therein). Many previous studies focus on the *nondifferentiability* of the regularization term. In contrast, we focus on the *couplings* between variables (or non-separability) caused by the design matrix. In fact, if the optimization problem can be decomposed into smaller (e.g., containing a single variable) problems, optimization is easy. Recently Wright et al. (2009) showed that the so called iterative shrinkage/thresholding (IST) method (see Figueiredo and Nowak, 2003; Daubechies et al., 2004; Combettes and Wajs, 2005; Figueiredo et al., 2007a) can be seen as an iterative *separable approximation* process.

In this paper, we show that a recently proposed dual augmented Lagrangian (DAL) algorithm (Tomioka and Sugiyama, 2009) can be considered as an *exact* (up to finite tolerance) version of the iterative approximation process discussed in Wright et al. (2009). Our formulation is based on the connection between the proximal minimization (Rockafellar, 1976a) and the augmented Lagrangian (AL) algorithm (Hestenes, 1969; Powell, 1969; Rockafellar, 1976b; Bertsekas, 1982). The proximal minimization framework also allows us to rigorously study the convergence behaviour of DAL. We show that DAL converges super-linearly under some mild conditions, which means that the number of iterations that we need to obtain an $\varepsilon$-accurate solution grows no greater than logarithmically with $1/\varepsilon$. Due to the generality of the framework, our analysis applies to a wide variety of practically important regularizers. Our analysis improves the classical result on the convergence of augmented Lagrangian algorithms in Rockafellar (1976b) by taking special structures of sparse estimation into account. In addition, we make no asymptotic arguments as in Rockafellar (1976b) and Kort and Bertsekas (1976); instead our convergence analysis is build on top of the recent result in Beck and Teboulle (2009).

Augmented Lagrangian formulations have also been considered in Yin et al. (2008) and Goldstein and Osher (2009) for sparse signal reconstruction. What differentiates DAL approach of Tomioka and Sugiyama (2009) from those studied earlier is that the AL algorithm is applied to the dual problem (see Section 2.2), which results in an inner minimization problem that can be solved efficiently exploiting the sparsity of intermediate solutions (see Section 4.1). Applying AL

| | primal loss $f_\ell(\mathbf{z})$ | conjugate loss $f_\ell^*(-\alpha)$ | gradient $-\nabla f_\ell^*(-\alpha)$ | Hessian $\nabla^2 f_\ell^*(-\alpha)$ | $\gamma$ |
|---|---|---|---|---|---|
| Squared loss | $\frac{1}{2}\sum_{i=1}^n \frac{(y_i-z_i)^2}{\sigma_i^2}$ | $\frac{1}{2}\sum_{i=1}^n \sigma_i^2(\alpha_i-y_i)^2$ | $\mathrm{diag}\,(\sigma_1^2,\ldots,\sigma_m^2)\,(\alpha-\mathbf{y})$ | $\mathrm{diag}\,(\sigma_1^2,\ldots,\sigma_m^2)$ | $\min_i \sigma_i^2$ |
| Logistic loss | $\sum_{i=1}^n \log(1+\exp(-y_i z_i))$ | $\sum_{i=1}^n \big((\alpha_i y_i)\log(\alpha_i y_i)$ $+(1-\alpha_i y_i)\log(1-\alpha_i y_i)\big)$ | $\Big(y_i \log \frac{\alpha_i y_i}{1-\alpha_i y_i}\Big)_{i=1}^m$ | $\mathrm{diag}(\frac{1}{\alpha_i y_i(1-\alpha_i y_i)})_{i=1}^m$ | 4 |
| Hyperbolic secant likelihood (Haufe et al., 2010) | $\sum_{i=1}^n \log\left(e^{y_i-z_i}+e^{-y_i+z_i}\right)$ | $\frac{1}{2}\sum_{i=1}^n \big((1-\alpha_i)\log(1-\alpha_i)$ $+(1+\alpha_i)\log(1+\alpha_i)-2\alpha_i y_i\big)$ | $(\frac{1}{2}\log\frac{1+\alpha_i}{1-\alpha_i}-y_i)_{i=1}^m$ | $\mathrm{diag}(\frac{1}{2(1-\alpha_i)(1+\alpha_i)})_{i=1}^m$ | 2 |
| Multi-class logit (Tomioka and Müller, 2010) | $\sum_{i=1}^m \big(-\sum_{k=1}^{c-1} z_{i(k)} y_{ik}$ $+\log\big(\sum_{k=1}^{c-1} e^{z_{i(k)}}+1\big)\big)$ | $\sum_{i=1}^m \big(\sum_{k=1}^{c-1}(y_{ik}-\alpha_{i(k)})\log(y_{ik}-\alpha_{i(k)})$ $+(y_{ic}+\sum_{k=1}^{c-1}\alpha_{i(k)})\log(y_{ic}+\sum_{k=1}^{c-1}\alpha_{i(k)})\big)$ $(0\le y_{ik}-\alpha_{i(k)}\le 1$ $(k=1,\ldots,c-1),$ $0\le y_{ic}+\sum_{k=1}^{c-1}\alpha_{i(k)}\le 1)$ | $\big(-\log\frac{y_{ik}-\alpha_{i(k)}}{y_{ic}+\sum_{k=1}^{c-1}\alpha_{i(k)}}\big)_{i(k)=1}^{m(c-1)}$ $(i=1,\ldots,m;$ $k=1,\ldots,c-1)$ | $\Big(\frac{\delta_{i,j}\delta_{k,l}}{y_{ik}-\alpha_{i(k)}}$ $+\frac{\delta_{i,j}}{y_{ic}+\sum_{k'=1}^{c-1}\alpha_{i(k')}}\Big)_{i(k),j(l)=1}^{m(c-1)}$ $(i,j=1,\ldots,m;$ $k,l=1,\ldots,c-1)$ | 1 |

Table 1: List of loss functions and their convex conjugates. Constant terms are ignored. For the multi-class logit loss, $\mathbf{z},\alpha\in\mathbb{R}^{m(c-1)}$, where $m$ is the number of samples and $c$ is the number of classes; $y_{ik}=1$ if the $i$th sample belongs to the $k$th class, and zero otherwise; $i(k):=(i-1)c+k$ denotes the linear index corresponding to the $k$th output for the $i$th sample; $\delta_{i,k}$ denotes the Kronecker delta function.

formulation to the dual problem also plays an important role in the convergence analysis because some loss functions (e.g., logistic loss) are not strongly convex in the primal; see Section 5. Recently Yang and Zhang (2009) compared primal and dual augmented Lagrangian algorithms for $\ell_1$-problems and reported that the dual formulation was more efficient. See also Tomioka et al. (2011b) for related discussions.

This paper is organized as follows. In Section 2, we mathematically formulate the sparse estimation problem and we review DAL algorithm. We derive DAL algorithm from the proximal minimization framework in Section 3; special instances of DAL algorithm are discussed in Section 4. In Section 5, we theoretically analyze the convergence behaviour of DAL algorithm. We discuss previously proposed algorithms in Section 6 and contrast them with DAL. In Section 7 we confirm our analysis in a simulated $\ell_1$-regularized logistic regression problem. Moreover, we extensively compare recently proposed algorithms for $\ell_1$-regularized logistic regression including DAL in synthetic and benchmark data sets under a variety of conditions. Finally we summarize our contribution in Section 8. Most of the proofs are given in the appendix.

## 2. Sparse Estimation Problem and DAL Algorithm

In this section, we first formulate the sparse estimation problem as a convex optimization problem, and state our assumptions. Next we derive DAL algorithm for $\ell_1$-problem as an augmented Lagrangian method in the dual.

### 2.1 Objective

We consider the problem of estimating an $n$ dimensional parameter vector from $m$ training examples as described in the following optimization problem:

$$\underset{\mathbf{w}\in\mathbb{R}^n}{\text{minimize}} \quad \underbrace{f_\ell(\mathbf{A}\mathbf{w}) + \phi_\lambda(\mathbf{w})}_{=:f(\mathbf{w})}, \tag{1}$$

where $\mathbf{w}\in\mathbb{R}^n$ is the parameter vector to be estimated, $\mathbf{A}\in\mathbb{R}^{m\times n}$ is a design matrix, and $f_\ell(\cdot)$ is a loss function. We call the first term in the minimand the loss term and the second term the regularization term, or the regularizer.

We assume that the loss function $f_\ell : \mathbb{R}^m \to \mathbb{R}\cup\{+\infty\}$ is a closed proper strictly convex function.[1] See Table 1 for examples of loss functions. We assume that $f_\ell$ has Lipschitz continuous gradient with modulus $1/\gamma$ (see Assumption **(A2)** in Section 5.2). If $f_\ell$ is twice differentiable, this condition is equivalent to saying that the maximum eigenvalue of the Hessian of $f_\ell$ is uniformly bounded by $1/\gamma$. Such $\gamma$ exists for example for quadratic loss, logistic loss, and other log-linear losses. However, non-smooth loss functions (e.g., the hinge loss and the absolute loss) are excluded. Note that since we separate the data matrix $\mathbf{A}$ from the loss function, we can quantify the above constant $\gamma$ without examining the data. Moreover, we assume that the convex conjugate[2] $f_\ell^*$ is (essentially) twice differentiable. Note that the first order differentiability of the convex conjugate $f_\ell^*$ is implied by the strict convexity of the loss function $f_\ell$ (Rockafellar, 1970, Theorem 26.3).

1. "Closed" means that the epigraph $\{(\mathbf{z},y)\in\mathbb{R}^{m+1} : y \geq f_\ell(\mathbf{z})\}$ is a closed set, and "proper" means that the function is not everywhere $+\infty$; see, for example, Rockafellar (1970). In the sequel, we use the word "convex function" in the meaning of "closed proper convex function".
2. The convex conjugate of a function $f:\mathbb{R}^n\to\mathbb{R}\cup\{+\infty\}$ is a function $f^*$ over $\mathbb{R}^n$ that takes values in $\mathbb{R}^n\cup\{+\infty\}$ and is defined as $f^*(\mathbf{y}) = \sup_{\mathbf{x}\in\mathbb{R}^n}(\mathbf{y}^\top\mathbf{x} - f(\mathbf{x}))$.

The regularization term $\phi_\lambda(\mathbf{w})$ is a convex possibly nondifferentiable function. In addition, we assume that for all $\eta > 0$, $\eta\phi_\lambda(\mathbf{w}) = \phi_{\eta\lambda}(\mathbf{w})$.

An important special case, which has been studied by many authors (Tibshirani, 1996; Efron et al., 2004; Andrew and Gao, 2007; Koh et al., 2007) is the $\ell_1$-regularization:

$$\underset{\mathbf{w}\in\mathbb{R}^n}{\text{minimize}} \quad f_\ell(\mathbf{A}\mathbf{w}) + \lambda\|\mathbf{w}\|_1, \tag{2}$$

where $\|\mathbf{w}\|_1 = \sum_{j=1}^n |w_j|$ is the $\ell_1$-norm of $\mathbf{w}$.

### 2.2 Dual Augmented Lagrangian (DAL) Algorithm

In this subsection, we review DAL algorithm following the line of Tomioka and Sugiyama (2009). Although, the squared loss function and the $\ell_1$-regularizer were considered in the original paper, we deal with a slightly more general setting in Equation (2) for notational convenience; that is, we consider general closed convex loss functions instead of the squared loss. For general information on augmented Lagrangian algorithms (Powell, 1969; Hestenes, 1969; Rockafellar, 1976b), see Bertsekas (1982) and Nocedal and Wright (1999).

Let $\phi_\lambda(\mathbf{w})$ be the $\ell_1$-regularizer, that is, $\phi_\lambda(\mathbf{w}) = \lambda\|\mathbf{w}\|_1 = \lambda\sum_{j=1}^n |w_j|$. Using the Fenchel duality theorem (Rockafellar, 1970), the dual of the problem (2) can be written as follows:

$$\underset{\alpha\in\mathbb{R}^m,\mathbf{v}\in\mathbb{R}^n}{\text{maximize}} \quad -f_\ell^*(-\alpha) - \delta_\lambda^\infty(\mathbf{v}), \tag{3}$$

$$\text{subject to} \quad \mathbf{v} = \mathbf{A}^\top\alpha, \tag{4}$$

where $\delta_\lambda^\infty$ is the indicator function (Rockafellar, 1970, p28) of the $\ell_\infty$-ball of radius $\lambda$, namely

$$\delta_\lambda^\infty(\mathbf{v}) = \sum_{j=1}^n \delta_\lambda^\infty(v_j), \tag{5}$$

where $\delta_\lambda^\infty(v_j) = 0$, if $|v_j| \le \lambda$, and $+\infty$ otherwise.

Let us consider the augmented Lagrangian (AL) function $L_\eta$ with respect to the above dual problem (3)

$$L_\eta(\alpha, \mathbf{v}; \mathbf{w}) = -f_\ell^*(-\alpha) - \delta_\lambda^\infty(\mathbf{v}) + \mathbf{w}^\top(\mathbf{v} - \mathbf{A}^\top\alpha) - \frac{\eta}{2}\|\mathbf{v} - \mathbf{A}^\top\alpha\|^2, \tag{6}$$

where the primal variable $\mathbf{w} \in \mathbb{R}^n$ is interpreted as a Lagrangian multiplier vector in the AL framework. Note that the AL function is the ordinary Lagrangian if $\eta = 0$.

Let $\eta_0, \eta_1, \ldots$ be a non-decreasing sequence of positive numbers. At every time step $t$, given the current primal solution $\mathbf{w}^t$, we maximize the AL function $L_{\eta_t}(\alpha, \mathbf{v}; \mathbf{w}^t)$ with respect to $\alpha$ and $\mathbf{v}$. The maximizer $(\alpha^t, \mathbf{v}^t)$ is used to update the primal solution (Lagrangian multiplier) $\mathbf{w}^t$ as follows:

$$\mathbf{w}^{t+1} = \mathbf{w}^t + \eta_t(\mathbf{A}^\top\alpha^t - \mathbf{v}^t). \tag{7}$$

Note that the maximization of the AL function (6) with respect to $\mathbf{v}$ can be carried out in a closed form, because the terms involved in the maximization can be separated into $n$ terms, each containing single $v_j$, as follows:

$$L_{\eta_t}(\alpha, \mathbf{v}) = -f_\ell^*(-\alpha) - \sum_{j=1}^n \left(\frac{\eta_t}{2}(v_j - (\mathbf{w}^t/\eta_t + \mathbf{A}^\top\alpha)_j)^2 + \delta_\lambda^\infty(v_j)\right),$$

where $(\cdot)_j$ denotes the $j$th element of a vector. Since $\delta_\lambda^\infty(v_j)$ is infinity outside the domain $-\lambda \leq v_j \leq \lambda$, the maximizer $\mathbf{v}^t(\alpha)$ is obtained as a projection onto the $\ell_\infty$ ball of radius $\lambda$ as follows (see also Figure 9):

$$\mathbf{v}^t(\alpha) = \mathrm{proj}_{[-\lambda,\lambda]}\left(\mathbf{w}^t/\eta_t + \mathbf{A}^\top\alpha\right) := \left(\min\left(|y_j|,\lambda\right)\frac{y_j}{|y_j|}\right)_{j=1}^n, \tag{8}$$

where $(y_j)_{j=1}^n$ denotes an $n$-dimensional vector whose $j$th element is given by $y_j$. Note that the ratio $y_j/|y_j|$ is defined to be zero[3] if $y_j = 0$. Substituting the above $\mathbf{v}^t$ back into Equation (7), we obtain the following update equation:

$$\mathbf{w}^{t+1} = \mathrm{prox}^{\ell_1}_{\lambda\eta_t}(\mathbf{w}^t + \eta_t\mathbf{A}^\top\alpha^t),$$

where $\mathrm{prox}^{\ell_1}_{\lambda\eta_t}$ is called the soft-threshold operation[4] and is defined as follows:

$$\mathrm{prox}^{\ell_1}_{\lambda}(\mathbf{y}) := \left(\max(|y_j| - \lambda, 0)\frac{y_j}{|y_j|}\right)_{j=1}^n. \tag{9}$$

The soft-threshold operation is well known in signal processing community and has been studied extensively (Donoho, 1995; Figueiredo and Nowak, 2003; Daubechies et al., 2004; Combettes and Wajs, 2005).

Furthermore, substituting the above $\mathbf{v}^t(\alpha)$ into Equation (6), we can express $\alpha^t$ as the minimizer of the function

$$\varphi_t(\alpha) := -L_{\eta_t}(\alpha, \mathbf{v}^t(\alpha); \mathbf{w}^t) = f_\ell^*(-\alpha) + \frac{1}{2\eta_t}\|\mathrm{prox}^{\ell_1}_{\lambda\eta_t}(\mathbf{w}^t + \eta_t\mathbf{A}^\top\alpha)\|^2, \tag{10}$$

which we also call an AL function with a slight abuse of terminology. Note that the maximization in Equation (6) is turned into a minimization of the above function by negating the AL function.

## 3. Proximal Minimization View

The first contribution of this paper is to derive DAL algorithm we reviewed in Section 2.2 from the proximal minimization framework (Rockafellar, 1976a), which allows for a new interpretation of the algorithm (see Section 3.3) and rigorous analysis of its convergence (see Section 5).

### 3.1 Proximal Minimization Algorithm

Let us consider the following iterative algorithm called the proximal minimization algorithm (Rockafellar, 1976a) for the minimization of the objective (1).

1. Choose some initial solution $\mathbf{w}^0$ and a sequence of non-decreasing positive numbers $\eta_0 \leq \eta_1 \leq \cdots$.

3. This is equivalent to defining $y_j/|y_j| = \mathrm{sign}(y_j)$. We use $y_j/|y_j|$ instead of $\mathrm{sign}(y_j)$ to define the soft-threshold operations corresponding to $\ell_1$ and the group-lasso regularizations (see Section 4.2) in a similar way.
4. This notation is a simplified version of the general notation we introduce later in Equation (15).

2. Repeat until some criterion (e.g., duality gap Wright et al., 2009; Tomioka and Sugiyama, 2009) is satisfied:

$$\mathbf{w}^{t+1} = \operatorname*{argmin}_{\mathbf{w}\in\mathbb{R}^n} \left( f(\mathbf{w}) + \frac{1}{2\eta_t}\|\mathbf{w}-\mathbf{w}^t\|^2 \right), \tag{11}$$

where $f(\mathbf{w})$ is the objective function in Equation (1) and $\eta_t$ controls the influence of the additional *proximity term*.

The proximity term tries to keep the next solution $\mathbf{w}^{t+1}$ close to the current solution $\mathbf{w}^t$. Importantly, the objective (11) is strongly convex even if the original objective (1) is not; see Rockafellar (1976b). Although at this point it is not clear how we are going to carry out the above minimization, by definition we have $f(\mathbf{w}^{t+1}) + \frac{1}{2\eta_t}\|\mathbf{w}^{t+1}-\mathbf{w}^t\|^2 \leq f(\mathbf{w}^t)$; that is, provided that the step-size is positive, the function value decreases monotonically at every iteration.

### 3.2 Iterative Shrinkage/Thresholding Algorithm from the Proximal Minimization Framework

The function to be minimized in Equation (11) is strongly convex. However, there seems to be no obvious way to minimize Equation (11), because it is still (possibly) nondifferentiable and cannot be decomposed into smaller problems because the elements of $\mathbf{w}$ are coupled.

One way to make the proximal minimization algorithm practical is to linearly approximate (see Wright et al., 2009) the loss term at the current point $\mathbf{w}^t$ as

$$f_\ell(\mathbf{A}\mathbf{w}) \simeq f_\ell(\mathbf{A}\mathbf{w}^t) + \left(\nabla f_\ell^t\right)^\top \mathbf{A}\left(\mathbf{w}-\mathbf{w}^t\right), \tag{12}$$

where $\nabla f_\ell^t$ is a short hand for $\nabla f_\ell(\mathbf{A}\mathbf{w}^t)$. Substituting the above approximation (12) into the iteration (11), we obtain

$$\mathbf{w}^{t+1} = \operatorname*{argmin}_{\mathbf{w}\in\mathbb{R}^n} \left( \left(\nabla f_\ell^t\right)^\top \mathbf{A}\mathbf{w} + \phi_\lambda(\mathbf{w}) + \frac{1}{2\eta_t}\|\mathbf{w}-\mathbf{w}^t\|^2 \right), \tag{13}$$

where constant terms are omitted from the right-hand side. Note that because of the linear approximation, there is no coupling between the elements of $\mathbf{w}$. For example, if $\phi_\lambda(\mathbf{w}) = \lambda\|\mathbf{w}\|$, the minimand in the right-hand side of the above equation can be separated into $n$ terms each containing single $w_j$, which can be separately minimized.

Rewriting the above update equation, we obtain the well-known iterative shrinkage/ thresholding (IST) method[5] (Figueiredo and Nowak, 2003; Daubechies et al., 2004; Combettes and Wajs, 2005; Figueiredo et al., 2007a). The IST iteration can be written as follows:

$$\mathbf{w}^{t+1} := \operatorname{prox}_{\phi_{\lambda\eta_t}}\left(\mathbf{w}^t - \eta_t \mathbf{A}^\top \nabla f_\ell^t\right), \tag{14}$$

where the proximity operator $\operatorname{prox}_{\phi_{\lambda\eta_t}}$ is defined as follows:

$$\operatorname{prox}_{\phi_\lambda}(\mathbf{y}) = \operatorname*{argmin}_{\mathbf{x}\in\mathbb{R}^n} \left( \frac{1}{2}\|\mathbf{y}-\mathbf{x}\|^2 + \phi_\lambda(\mathbf{x}) \right). \tag{15}$$

Note that the soft-threshold operation $\operatorname{prox}_\lambda^{\ell_1}$ (9) is the proximity operator corresponding to the $\ell_1$-regularizer $\phi_\lambda^{\ell_1}(\mathbf{w}) = \lambda\|\mathbf{w}\|_1$.

5. It is also known as the forward-backward splitting method (Lions and Mercier, 1979; Combettes and Wajs, 2005; Duchi and Singer, 2009); see Section 6.

### 3.3 DAL Algorithm from the Proximal Minimization Framework

The above IST approach can be considered to be constructing a linear lower bound of the loss term in Equation (11) at the *current point* $\mathbf{w}^t$. In this subsection we show that we can precisely (to finite precision) minimize Equation (11) using a *parametrized* linear lower bound that can be adjusted to be the tightest at the *next point* $\mathbf{w}^{t+1}$. Our approach is based on the convexity of the loss function $f_\ell$. First note that we can rewrite the loss function $f_\ell$ as a point-wise maximum as follows:

$$f_\ell(\mathbf{A}\mathbf{w}) = \max_{\alpha\in\mathbb{R}^m} \left( (-\alpha)^\top \mathbf{A}\mathbf{w} - f_\ell^*(-\alpha) \right), \tag{16}$$

where $f_\ell^*$ is the convex conjugate functions of $f_\ell$. Now we substitute this expression into the iteration (11) as follows:

$$\mathbf{w}^{t+1} = \operatorname*{argmin}_{\mathbf{w}\in\mathbb{R}^n} \max_{\alpha\in\mathbb{R}^m} \left\{ -\alpha^\top \mathbf{A}\mathbf{w} - f_\ell^*(-\alpha) + \phi_\lambda(\mathbf{w}) + \frac{1}{2\eta_t}\|\mathbf{w}-\mathbf{w}^t\|^2 \right\}. \tag{17}$$

Note that now the loss term is expressed as a *linear* function as in the IST approach; see Equation (13). Now we exchange the order of minimization and maximization because the function to be minimaxed in Equation (17) is a saddle function (i.e., convex with respect to $\mathbf{w}$ and concave with respect to $\alpha$ Rockafellar, 1970), as follows:

$$\begin{aligned} &\min_{\mathbf{w}\in\mathbb{R}^n} \max_{\alpha\in\mathbb{R}^m} \left\{ -\alpha^\top \mathbf{A}\mathbf{w} - f_\ell^*(-\alpha) + \phi_\lambda(\mathbf{w}) + \frac{1}{2\eta_t}\|\mathbf{w}-\mathbf{w}^t\|^2 \right\} \\ &\quad = \max_{\alpha\in\mathbb{R}^m} \left\{ -f_\ell^*(-\alpha) + \min_{\mathbf{w}\in\mathbb{R}^n} \left( -\alpha^\top \mathbf{A}\mathbf{w} + \phi_\lambda(\mathbf{w}) + \frac{1}{2\eta_t}\|\mathbf{w}-\mathbf{w}^t\|^2 \right) \right\}. \end{aligned} \tag{18}$$

Notice the similarity between the two minimizations (13) and (18) (with fixed $\alpha$).

The minimization with respect to $\mathbf{w}$ in Equation (18) gives the following update equation

$$\mathbf{w}^{t+1} = \operatorname{prox}_{\phi_{\lambda\eta_t}} \left( \mathbf{w}^t + \eta_t \mathbf{A}^\top \alpha^t \right), \tag{19}$$

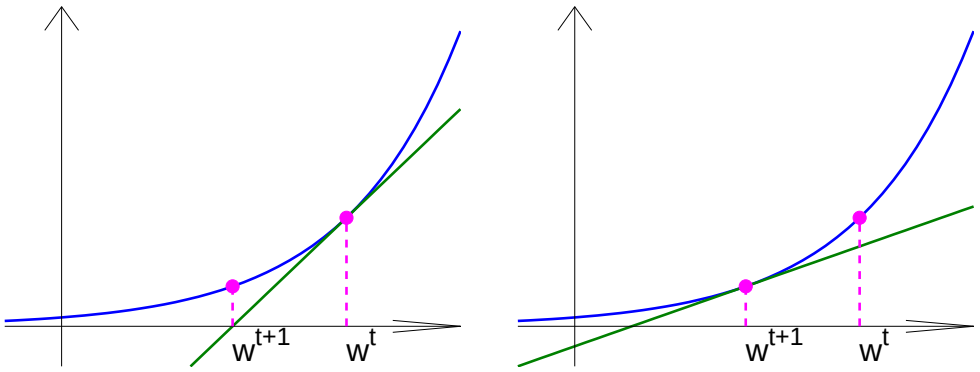
where $\alpha^t$ denotes the maximizer with respect to $\alpha$ in Equation (18). Note that $\alpha^t$ is in general different from $-\nabla f_\ell^t$ used in the IST approach (14). Actually, we show below that $\alpha^t = -\nabla f_\ell^{t+1}$ if the max-min problem (18) is solved exactly. Therefore taking $\alpha^t = -\nabla f_\ell^t$ can be considered as a naive approximation to this.

The final step to derive DAL algorithm is to compute the maximizer $\alpha^t$ in Equation (18). This step is slightly involved and the derivation is presented in Appendix B. The result of the derivation can be written as follows (notice that the maximization in Equation (18) is turned into a minimization by reversing the sign):

$$\alpha^t = \operatorname*{argmin}_{\alpha\in\mathbb{R}^m} \underbrace{\left( f_\ell^*(-\alpha) + \frac{1}{\eta_t}\Phi^*_{\lambda\eta_t}(\mathbf{w}^t + \eta_t \mathbf{A}^\top \alpha) \right)}_{=:\varphi_t(\alpha)}, \tag{20}$$

where the function $\Phi^*_{\lambda\eta_t}$ is called the Moreau envelope of $\phi^*_\lambda$ (see Moreau, 1965; Rockafellar, 1970) and is defined as follows:

$$\Phi^*_\lambda(\mathbf{w}) = \min_{\mathbf{x}\in\mathbb{R}^n} \left( \phi^*_\lambda(\mathbf{x}) + \frac{1}{2}\|\mathbf{x}-\mathbf{w}\|^2 \right). \tag{21}$$




(a) Gradient-based lower-bound used in IST (b) Variational lower-bound used in DAL

Figure 1: Comparison of the lower bounds used in IST and DAL.

See Appendix A for more details. Since the function $\varphi_t(\alpha)$ in Equation (20) generalizes the AL-function (10), we call it an augmented Lagrangian (AL) function.

What we need to do at every iteration is to minimize the AL function $\varphi_t(\alpha)$ and update the Lagrangian multiplier $\mathbf{w}^t$ as in Equation (19) using the minimizer $\alpha^t$ in Equation (20). Of course in practice we would like to stop the inner minimization at a finite tolerance. We discuss the stopping condition in Section 5.

The algorithm we derived above is indeed a generalization of DAL algorithm we reviewed in Section 2.2. This can be shown by computing the proximity operator (19) and the Moreau envelope (21) for the specific case of $\ell_1$-regularization; see Section 4.1 and also Table 2.

The AL function $\varphi_t(\alpha)$ is continuously differentiable, because the AL function is a sum of $f_\ell^*$ (differentiable by assumption) and an envelope function (differentiable; see Appendix A). In fact, using Lemma 10 in Appendix A, the derivative of the AL function can be evaluated as follows:

$$\nabla\varphi_t(\alpha) = -\nabla f_\ell^*(-\alpha) + \mathbf{A}\mathbf{w}^{t+1}(\alpha), \tag{22}$$

where $\mathbf{w}^{t+1}(\alpha) := \mathrm{prox}_{\phi_\lambda \eta_t}\left(\mathbf{w}^t + \eta_t \mathbf{A}^\top \alpha\right)$. The expression for the second derivative depends on the particular regularizer chosen.

Notice again that the above update Equation (19) is very similar to the one in the IST approach Equation (14). However, $-\alpha$, which is the slope of the lower-bound (16) is optimized in the inner minimization (20) so that the lower-bound is the tightest at the *next point* $\mathbf{w}^{t+1}$. In fact, if $\nabla\varphi_t(\alpha) = 0$ then $\nabla f_\ell(\mathbf{A}\mathbf{w}^{t+1}) = -\alpha^t$ because of Equation (22) and $\nabla f_\ell(\nabla f_\ell^*(-\alpha^t)) = -\alpha^t$. The difference between the strategies used in IST and DAL to construct a lower-bound is highlighted in Figure 1. IST uses a fixed gradient-based lower-bound which is tightest at the current solution $\mathbf{w}^t$, whereas DAL uses a variational lower-bound, which can be adjusted to become tightest at the next solution $\mathbf{w}^{t+1}$.

The general connection between the augmented Lagrangian algorithm and the proximal minimization algorithm, and (asymptotic) convergence results can be found in Rockafellar (1976b) and Bertsekas (1982). The derivation we show above is a special case when the objective function $f(\mathbf{w})$ can be split into a part that is easy to handle (regularization term $\phi_\lambda(\mathbf{w})$) and the rest (loss term $f_\ell(\mathbf{A}\mathbf{w})$).

| Description | Regularizer | Proximity operator $\mathrm{prox}_\lambda$ | Envelope function $\Phi^*_\lambda$ |
|---|---|---|---|
| $\ell_1$-regularizer (Tibshirani, 1996) | $\phi^{\ell_1}_\lambda(\mathbf{w}) = \lambda\sum_{j=1}^n \lvert w_j\rvert$ | $\mathrm{prox}^{\ell_1}_\lambda(\mathbf{w}) = \left((\lvert w_j\rvert-\lambda)_+\frac{w_j}{\lvert w_j\rvert}\right)_{j=1}^n$ | $\Phi^*_\lambda(\mathbf{w}) = \frac{1}{2}\sum_{j=1}^n(\lvert w_j\rvert-\lambda)_+^2$ |
| Group lasso (Yuan and Lin, 2006) | $\phi^{\mathsf{G}}_\lambda(\mathbf{w}) = \lambda\sum_{\mathsf{g}\in\mathsf{G}}\lVert\mathbf{w}_\mathsf{g}\rVert$ | $\mathrm{prox}^{\mathsf{G}}_\lambda(\mathbf{w}) = \left((\lVert\mathbf{w}_\mathsf{g}\rVert-\lambda)_+\frac{\mathbf{w}_\mathsf{g}}{\lVert\mathbf{w}_\mathsf{g}\rVert}\right)_{\mathsf{g}\in\mathsf{G}}$ | $\Phi^*_\lambda(\mathbf{w}) = \frac{1}{2}\sum_{\mathsf{g}\in\mathsf{G}}(\lVert\mathbf{w}_\mathsf{g}\rVert-\lambda)_+^2$ |
| Trace norm (Fazel et al., 2001; Srebro et al., 2005; Tomioka et al., 2010) | $\phi^{\mathrm{tr}}_\lambda(\mathbf{w}) = \lambda\sum_{j=1}^n\sigma_j(\mathbf{w})$ | $\mathrm{prox}^{\mathrm{tr}}_\lambda(\mathbf{w}) = \mathrm{vec}\left(\mathbf{U}(\mathbf{S}-\lambda)_+\mathbf{V}^\top\right)$ | $\Phi^*_\lambda(\mathbf{w}) = \frac{1}{2}\sum_{j=1}^r(\sigma_j(\mathbf{w})-\lambda)_+^2$ |
| Elastic-net (Zou and Hastie, 2005; Tomioka and Suzuki, 2010) | $\phi^{\mathrm{en}}_\lambda(\mathbf{w}) = \lambda\sum_{j=1}^n((1-\theta)\lvert w_j\rvert+\frac{\theta}{2}w_j^2)$ | $\mathrm{prox}^{\mathrm{en}}_\lambda(\mathbf{w}) = \left(\frac{(\lvert w_j\rvert-\lambda(1-\theta))_+}{1+\lambda\theta}\frac{w_j}{\lvert w_j\rvert}\right)_{j=1}^n$ | $\Phi^*_\lambda(\mathbf{w}) = \frac{1+\lambda\theta}{2}\sum_{j=1}^n\left(\frac{\lvert w_j\rvert-\lambda(1-\theta)}{1+\lambda\theta}\right)_+^2$ |

Table 2: List of regularizers and their corresponding proximity operators (15) and the Envelope function (21). The operation $(\cdot)_+$ is defined as $(x)_+ := \max(0,x)$ and applies element-wise to a matrix.

**Algorithm 1** DAL algorithm for $\ell_1$-regularization

1: **Input:** design matrix $\mathbf{A}$, loss function $f_\ell$, regularization constant $\lambda$, sequence of proximity parameters $\eta_t$ ($t = 0, 1, 2, \ldots$), initial solution $\mathbf{w}^0$, tolerance $\varepsilon$.
2: Set $t = 0$.
3: **repeat**
4: Minimize the augmented Lagrangian function $\varphi_t(\alpha)$ (see Equation 10) with the gradient and Hessian given in Equations (24) and (25), respectively, using Newton's method. Let $\alpha^t$ be the approximate minimizer

$$\alpha^t \simeq \operatorname*{argmin}_{\alpha \in \mathbb{R}^m} \Big( f_\ell^*(-\alpha) + \frac{1}{2\eta_t} \left\| \operatorname{prox}^{\ell_1}_{\lambda\eta_t}(\mathbf{w}^t + \eta_t \mathbf{A}^\top \alpha) \right\|^2 \Big),$$

with the stopping criterion (see Section 5.2)

$$\|\nabla \varphi_t(\alpha^t)\| \le \sqrt{\frac{\gamma}{\eta_t}} \left\| \operatorname{prox}^{\ell_1}_{\lambda\eta_t}(\mathbf{w}^t + \eta_t \mathbf{A}^\top \alpha^t) - \mathbf{w}^t \right\|,$$

where $\varphi_t(\alpha)$ is the derivative of the inner objective (24) and $1/\gamma$ is the Lipschitz constant of $\nabla f_\ell$.
5: Update $\mathbf{w}^{t+1} := \operatorname{prox}^{\ell_1}_{\lambda\eta_t}(\mathbf{w}^t + \eta_t \mathbf{A}^\top \alpha^t)$, $t \leftarrow t + 1$.
6: **until** relative duality gap (see Section 7.1.2) is less than the tolerance $\varepsilon$.
7: **Output:** the final solution $\mathbf{w}^t$.

## 4. Exemplary Instances

In this section, we discuss special instances of DAL framework presented in Section 3 and qualitatively discuss the efficiency of minimizing the inner objective. We first discuss the simple case of $\ell_1$-regularization (Section 4.1), and then group-lasso (Section 4.2) and other more general regularization using the so-called support functions (Section 4.3). In addition, the case of component-wise regularization is discussed in Section 4.4. See also Table 2 for a list of regularizers.

### 4.1 Dual Augmented Lagrangian Algorithm for $\ell_1$-Regularization

For the $\ell_1$-regularization, $\phi_\lambda^{\ell_1}(\mathbf{w}) = \lambda \|\mathbf{w}\|_1$, the update Equation (19) can be rewritten as follows:

$$\mathbf{w}^{t+1} = \operatorname{prox}^{\ell_1}_{\lambda\eta_t} \left( \mathbf{w}^t + \eta_t \mathbf{A}^\top \alpha^t \right), \tag{23}$$

where $\operatorname{prox}^{\ell_1}_\lambda$ is the proximity operator corresponding to the $\ell_1$-regularizer defined in Equation (9). Moreover, noticing that the convex conjugate of the $\ell_1$-regularizer is the indicator function $\delta^\infty_\lambda$ in Equation (5), we can derive the envelope function $\Phi^*_\lambda$ in Equation (21) as follows (see also Figure 9):

$$\Phi^*_\lambda(\mathbf{w}) = \frac{1}{2} \left\| \operatorname{prox}^{\ell_1}_\lambda(\mathbf{w}) \right\|^2.$$

Therefore, the AL function (10) in Tomioka and Sugiyama (2009) is derived from the proximal minimization framework (see Equation 20) in Section 3.

We use Newton's method for the minimization of the inner objective $\varphi_t(\alpha)$. The overall algorithm is shown in Algorithm 1. The gradient and Hessian of the AL function (10) can be evaluated

as follows (Tomioka and Sugiyama, 2009):

$$\nabla\varphi_t(\alpha) = -\nabla f_\ell^*(-\alpha) + \mathbf{A}\mathbf{w}^{t+1}(\alpha), \tag{24}$$

$$\nabla^2\varphi_t(\alpha) = \nabla^2 f_\ell^*(-\alpha) + \eta_t \mathbf{A}_+ \mathbf{A}_+{}^\top, \tag{25}$$

where $\mathbf{w}^{t+1}(\alpha) := \mathrm{prox}^{\ell_1}_{\lambda\eta_t}(\mathbf{w}^t + \eta_t \mathbf{A}^\top\alpha)$, and $\mathbf{A}_+$ is the matrix that consists of columns of $\mathbf{A}$ that corresponds to "active" variables (i.e., the non-zero elements of $\mathbf{w}^{t+1}(\alpha)$ ). Note that Equation (24) equals the general expression (22) from the proximal minimization framework.

It is worth noting that in both the computation of matrix-vector product in Equation (24) and the computation of matrix-matrix product in Equation (25), the cost is only proportional to the number of non-zero elements of $\mathbf{w}^{t+1}(\alpha)$. Thus when we are aiming for a sparse solution, the minimization of the AL function (10) can be performed efficiently.

### 4.2 Group Lasso

Let $\phi_\lambda$ be the group-lasso penalty (Yuan and Lin, 2006), that is,

$$\phi_\lambda^{\mathsf{G}}(\mathbf{w}) = \lambda \sum_{\mathsf{g}\in\mathsf{G}} \|\mathbf{w}_\mathsf{g}\|, \tag{26}$$

where $\mathsf{G}$ is a disjoint partition of the index set $\{1,\ldots,n\}$, and $\mathbf{w}_\mathsf{g} \in \mathbb{R}^{|\mathsf{g}|}$ is a sub-vector of $\mathbf{w}$ that consists of rows of $\mathbf{w}$ indicated by $\mathsf{g} \subseteq \{1,\ldots,n\}$. The proximity operator corresponding to the group-lasso regularizer $\phi_\lambda^{\mathsf{G}}$ is obtained as follows:

$$\mathrm{prox}_\lambda^{\mathsf{G}}(\mathbf{y}) := \mathrm{prox}_{\phi_\lambda^{\mathsf{G}}}(\mathbf{y}) = \left(\max(\|\mathbf{y}_\mathsf{g}\| - \lambda, 0)\frac{\mathbf{y}_\mathsf{g}}{\|\mathbf{y}_\mathsf{g}\|}\right)_{\mathsf{g}\in\mathsf{G}}, \tag{27}$$

where similarly to Equation (9), $(\mathbf{y}_\mathsf{g})_{\mathsf{g}\in\mathsf{G}}$ denotes an $n$-dimensional vector whose $\mathsf{g}$ component is given by $\mathbf{y}_\mathsf{g}$. Moreover, analogous to update Equation (23) (see also Equation 10) in the $\ell_1$-case, the update equations can be written as follows:

$$\mathbf{w}^{t+1} = \mathrm{prox}^{\mathsf{G}}_{\lambda\eta_t}\left(\mathbf{w}^t + \eta_t \mathbf{A}^\top\alpha^t\right),$$

where $\alpha^t$ is the minimizer of the AL function

$$\varphi_t(\alpha) = f_\ell^*(-\alpha) + \frac{1}{2\eta_t}\|\mathrm{prox}^{\mathsf{G}}_{\lambda\eta_t}(\mathbf{w}^t + \eta_t\mathbf{A}^\top\alpha)\|^2. \tag{28}$$

The overall algorithm is obtained by replacing the soft-thresholding operations in Algorithm 1 by the one defined above (27). In addition, the gradient and Hessian of the AL function $\varphi_t(\alpha)$ can be written as follows:

$$\nabla\varphi_t(\alpha) = -\nabla f_\ell^*(-\alpha) + \mathbf{A}\mathbf{w}^{t+1}(\alpha), \tag{29}$$

$$\nabla^2\varphi_t(\alpha) = \nabla^2 f_\ell^*(-\alpha) + \eta_t \sum_{\mathsf{g}\in\mathsf{G}^+} \mathbf{A}_\mathsf{g}\left(\left(1 - \frac{\lambda\eta_t}{\|\mathbf{q}_\mathsf{g}\|}\right)\mathbf{I}_{|\mathsf{g}|} + \frac{\lambda\eta_t}{\|\mathbf{q}_\mathsf{g}\|}\tilde{\mathbf{q}}_\mathsf{g}\tilde{\mathbf{q}}_\mathsf{g}{}^\top\right)\mathbf{A}_\mathsf{g}{}^\top, \tag{30}$$

where $\mathbf{w}^{t+1}(\alpha) = \mathrm{prox}^{\mathsf{G}}_{\lambda\eta_t}(\mathbf{w}^t + \eta_t \mathbf{A}^\top \alpha)$, and $\mathsf{G}^+$ is a subset of $\mathsf{G}$ that consists of active groups, namely $\mathsf{G}^+ := \{\mathsf{g} \in \mathsf{G} : \|\mathbf{w}^{t+1}_{\mathsf{g}}(\alpha)\| > 0\}$; $\mathbf{A}_{\mathsf{g}}$ is a sub-matrix of $\mathbf{A}$ that consists of columns corresponding to the index-set $\mathsf{g}$; $\mathbf{I}_{|\mathsf{g}|}$ is the $|\mathsf{g}| \times |\mathsf{g}|$ identity matrix; the vector $\mathbf{q} \in \mathbb{R}^n$ is defined as $\mathbf{q} := \mathbf{w}^t + \eta_t \mathbf{A}^\top \alpha$ and $\tilde{\mathbf{q}}_{\mathsf{g}} := \mathbf{q}_{\mathsf{g}} / \|\mathbf{q}_{\mathsf{g}}\|$, where $\mathbf{q}_{\mathsf{g}}$ is defined analogously to $\mathbf{w}_{\mathsf{g}}$. Note that in the above expression, $\lambda\eta_t / \|\mathbf{q}_{\mathsf{g}}\| \leq 1$ for $\mathsf{g} \in \mathsf{G}^+$ by the soft-threshold operation (27).

Similarly to the $\ell_1$-case in the last subsection, the sparsity of $\mathbf{w}^{t+1}(\alpha)$ (i.e., $|\mathsf{G}^+| \ll |\mathsf{G}|$) can be exploited to efficiently compute the gradient (29) and the Hessian (30).

### 4.3 Support Functions

The $\ell_1$-norm regularization and the group lasso regularization in Equation (26) can be generalized to the class of support functions. The support function of a convex set $C_\lambda$ is defined as follows:

$$\phi_\lambda(\mathbf{x}) = \sup_{\mathbf{y} \in C_\lambda} \mathbf{x}^\top \mathbf{y}. \tag{31}$$

For example, the $\ell_1$-norm is the support function of the $\ell_\infty$ unit ball (see Rockafellar, 1970) and the group lasso regularizer (26) is the support function of the group-generalized $\ell_\infty$-ball defined as $\{\mathbf{y} \in \mathbb{R}^n : \|\mathbf{y}_{\mathsf{g}}\| \leq \lambda, \forall \mathsf{g} \in \mathsf{G}\}$. It is well known that the convex conjugate of the support function (31) is the indicator function of $C$ (see Rockafellar, 1970), namely,

$$\phi^*_\lambda(\mathbf{y}) = \begin{cases} 0 & (\text{if } \mathbf{y} \in C_\lambda), \\ +\infty & (\text{otherwise}). \end{cases} \tag{32}$$

The proximity operator corresponding to the support function (31) can be written as follows:

$$\mathrm{prox}^{\mathrm{sup}}_{C_\lambda}(\mathbf{y}) := \mathbf{y} - \mathrm{proj}_{C_\lambda}(\mathbf{y}),$$

where $\mathrm{proj}_{C_\lambda}$ is the projection onto $C_\lambda$; see Lemma 8 in Appendix A. Finally, by computing the Moreau envelope (21) corresponding to the above $\phi^*_\lambda$, we have

$$\varphi_t(\alpha) = f^*_\ell(-\alpha) + \frac{1}{2\eta_t} \|\mathrm{prox}^{\mathrm{sup}}_{C_{\lambda\eta_t}}(\mathbf{w}^t + \eta_t \mathbf{A}^\top \alpha)\|^2, \tag{33}$$

where we used the fact that for the indicator function in Equation (32), $\phi^*_\lambda(\mathrm{proj}_{C_\lambda}(\mathbf{z})) = 0$ ($\forall \mathbf{z}$) and Lemma 8. Note that $C_\lambda = \{\mathbf{y} \in \mathbb{R}^n : \|\mathbf{y}\|_\infty \leq \lambda\}$ gives $\mathrm{prox}^{\mathrm{sup}}_{C_\lambda} = \mathrm{prox}^{\ell_1}_\lambda$ (see Equation 10), and $C_\lambda = \{\mathbf{y} \in \mathbb{R}^n : \|\mathbf{y}_{\mathsf{g}}\| \leq \lambda, \forall \mathsf{g} \in \mathsf{G}\}$ gives $\mathrm{prox}^{\mathrm{sup}}_{C_\lambda} = \mathrm{prox}^{\mathsf{G}}_\lambda$ (see Equation (28).

### 4.4 Handling Different Regularization Constant for Each Component

The $\ell_1$-regularizer in Section 4.1 and the group lasso regularizer in Section 4.2 assume that all the components (variables or groups) are regularized by the same constant $\lambda$. However the general formulation in Section 3.3 allows using different regularization constant for each component.

For example, let us consider the following regularizer:

$$\phi_\lambda(\mathbf{w}) = \sum_{j=1}^n \lambda_j |w_j|, \tag{34}$$

where $\lambda_j \geq 0$ ($j = 1, \ldots, n$). Note that we can also include unregularized terms (e.g., a bias term) by setting the corresponding regularization constant $\lambda_j = 0$. The soft-thresholding operation corresponding to the regularizer (34) is written as follows:

$$\mathrm{prox}_{\lambda}^{\ell_1}(\mathbf{y}) = \left( \max(|y_j| - \lambda_j, 0) \frac{y_j}{|y_j|} \right)_{j=1}^{n},$$

where again the ratio $y_j/|y_j|$ is defined to be zero if $y_j = 0$. Note that if $\lambda_j = 0$, the soft-thresholding operation is an identity mapping for that component. Moreover, by noticing that the regularizer (34) is a support function (see Section 4.3), the envelope function $\Phi_{\lambda}^*$ in Equation (21) is written as follows:

$$\Phi_{\lambda}^*(\mathbf{w}) = \frac{1}{2} \sum_{j=1}^{n} \max^2(|w_j| - \lambda_j, 0),$$

which can also be derived by noticing that $\Phi_{\lambda}^*(0) = 0$ and $\nabla \Phi_{\lambda}^*(\mathbf{y}) = \mathrm{prox}_{\lambda}^{\ell_1}(\mathbf{y})$ (Lemma 10 in Appendix A).

As a concrete example, let $b$ be an unregularized bias term and let us assume that all the components of $\mathbf{w} \in \mathbb{R}^n$ are regularized by the same regularization constant $\lambda$. In other words, we aim to solve the following optimization problem:

$$\underset{\mathbf{w} \in \mathbb{R}^n, b \in \mathbb{R}}{\text{minimize}} \quad f_{\ell}(\mathbf{A}\mathbf{w} + \mathbf{1}_m b) + \lambda \|\mathbf{w}\|_1,$$

where $\|\mathbf{w}\|_1$ is the $\ell_1$-norm of $\mathbf{w}$, and $\mathbf{1}_m$ is an $m$-dimensional all one vector. The update Equations (19) and (20) can be written as follows:

$$\mathbf{w}^{t+1} = \mathrm{prox}_{\lambda\eta_t}^{\ell_1}(\mathbf{w}^t + \eta_t \mathbf{A}^\top \alpha^t), \tag{35}$$

$$b^{t+1} = b^t + \eta_t \mathbf{1}_m{}^\top \alpha^t, \tag{36}$$

where $\alpha^t$ is the minimizer of the AL function as follows:

$$\alpha^t = \underset{\alpha \in \mathbb{R}^m}{\mathrm{argmin}} \left( f_{\ell}^*(-\alpha) + \frac{1}{2\eta_t} \left( \|\mathrm{prox}_{\lambda\eta_t}^{\ell_1}(\mathbf{w}^t + \eta_t \mathbf{A}^\top \alpha)\|^2 + (b^t + \eta_t \mathbf{1}_m{}^\top \alpha)^2 \right) \right). \tag{37}$$

## 5. Analysis

In this section, we first show the convergence of DAL algorithm assuming that the inner minimization problem (20) is solved exactly (Section 5.1), which is equivalent to the proximal minimization algorithm (11). The convergence is presented both in terms of the function value and the norm of the residual. Next, since it is impractical to perform the inner minimization to high precision, the finite tolerance version of the two theorems are presented in Section 5.2. The convergence rate obtained in Section 5.2 is slightly worse than the exact case. In Section 5.3, we show that the convergence rate can be improved by performing the inner minimization more precisely. Most of the proofs are given in Appendix C for the sake of readability.

Our result is inspired partly by Beck and Teboulle (2009) and is similar to the one given in Rockafellar (1976a) and Kort and Bertsekas (1976). However, our analysis does not require asymptotic arguments as in Rockafellar (1976a) or rely on the strong convexity of the objective as in Kort

and Bertsekas (1976). Importantly the stopping criterion we discuss in Section 5.2 can be checked in practice. Key to our analysis is the Lipschitz continuity of the gradient of the loss function $\nabla f_\ell$ and the assumption that the proximation with respect to $\phi_\lambda$ (see Equation 15) can be computed exactly. Connections between our assumption and the ones made in earlier studies are discussed in Section 5.4.

### 5.1 Exact Inner Minimization

**Lemma 1 (Beck and Teboulle, 2009)** *Let $\mathbf{w}^1, \mathbf{w}^2, \ldots$ be the sequence generated by the proximal minimization algorithm (Equation 11). For arbitrary $\mathbf{w} \in \mathbb{R}^n$ we have*

$$\eta_t(f(\mathbf{w}^{t+1}) - f(\mathbf{w})) \leq \frac{1}{2}\|\mathbf{w}^t - \mathbf{w}\|^2 - \frac{1}{2}\|\mathbf{w}^{t+1} - \mathbf{w}\|^2. \tag{38}$$

**Proof** First notice that $(\mathbf{w}^t - \mathbf{w}^{t+1})/\eta_t \in \partial f(\mathbf{w}^{t+1})$ because $\mathbf{w}^{t+1}$ minimizes Equation (11). Therefore using the convexity of $f$, we have[6]

$$\begin{aligned}
\eta_t(f(\mathbf{w}) - f(\mathbf{w}^{t+1})) &\geq \left\langle \mathbf{w} - \mathbf{w}^{t+1}, \mathbf{w}^t - \mathbf{w}^{t+1} \right\rangle \\
&= \left\langle \mathbf{w} - \mathbf{w}^{t+1}, \mathbf{w}^t - \mathbf{w} + \mathbf{w} - \mathbf{w}^{t+1} \right\rangle \\
&\geq \|\mathbf{w} - \mathbf{w}^{t+1}\|^2 - \|\mathbf{w} - \mathbf{w}^{t+1}\|\|\mathbf{w}^t - \mathbf{w}\| \\
&\geq \frac{1}{2}\|\mathbf{w} - \mathbf{w}^{t+1}\|^2 - \frac{1}{2}\|\mathbf{w}^t - \mathbf{w}\|^2,
\end{aligned} \tag{39}$$

where the third line follows from Cauchy-Schwartz inequality and the last line follows from the inequality of arithmetic and geometric means. ■

Note that DAL algorithm (Equations 19 and 20) with exact inner minimization generates a sequence from the proximal minimization algorithm (Equation 11). Therefore we have the following theorem.

**Theorem 2** *Let $\mathbf{w}^1, \mathbf{w}^2, \ldots$ be the sequence generated by DAL algorithm (Equations 19 and 20); let $W^*$ be the set of minimizers of the objective* (1) *and let $f(W^*)$ denote the minimum objective value. If the inner minimization (Equation 20) is solved exactly and the proximity parameter $\eta_t$ is increased exponentially, then the residual function value obtained by the DAL algorithm converges exponentially fast to zero as follows:*

$$f(\mathbf{w}^{k+1}) - f(W^*) \leq \frac{\|\mathbf{w}^0 - W^*\|^2}{2C_k}, \tag{40}$$

*where $\|\mathbf{w}^0 - W^*\|$ denotes the minimum distance between the initial solution $\mathbf{w}^0$ and $W^*$, namely, $\|\mathbf{w}^0 - W^*\| = \min_{\mathbf{w}^* \in W^*} \|\mathbf{w}^0 - \mathbf{w}^*\|$. Note that $C_k = \sum_{t=0}^k \eta_t$ also grows exponentially.*

6. We use the notation $\langle \mathbf{x}, \mathbf{y} \rangle := \sum_{j=1}^n x_j y_j$ for $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$.

**Proof** Let $\mathbf{w}^*$ be any point in $W^*$. Substituting $\mathbf{w} = \mathbf{w}^*$ in Equation (38) and summing both sides from $t = 1$ to $t = k$, we have

$$\left(\sum_{t=0}^{k} \eta_t\right) \left( \sum_{t=0}^{k} \frac{\eta_t f(\mathbf{w}^{t+1})}{\sum_{t=0}^{k} \eta_t} - f(\mathbf{w}^*) \right) \leq \frac{1}{2}\|\mathbf{w}^0 - \mathbf{w}^*\|^2 - \frac{1}{2}\|\mathbf{w}^{k+1} - \mathbf{w}^*\|^2 \leq \frac{1}{2}\|\mathbf{w}^0 - \mathbf{w}^*\|^2.$$

In addition, since $f(\mathbf{w}^{t+1}) \leq f(\mathbf{w}^t)$ $(t = 0, 1, 2, \ldots)$ from Equation (11), we have

$$\left(\sum_{t=0}^{k} \eta_t\right) \Big( f(\mathbf{w}^{k+1}) - f(\mathbf{w}^*) \Big) \leq \frac{1}{2}\|\mathbf{w}^0 - \mathbf{w}^*\|^2.$$

Finally, taking the minimum of the right-hand side with respect to $\mathbf{w}^* \in W^*$ and using the equivalence of proximal minimization (11) and DAL algorithm (19)-(20) (see Section 3.3), we complete the proof. ■

The above theorem claims the convergence of the residual function values $f(\mathbf{w}^t) - f(\mathbf{w}^*)$ obtained along the sequence $\mathbf{x}_1, \mathbf{x}_2, \ldots$. We can convert the above result into convergence in terms of the residual norm $\|\mathbf{w}^t - \mathbf{w}^*\|$ by introducing an assumption that connects the residual function value to the residual norm. In addition, we slightly generalize Lemma 1 to improve the convergence rate. Consequently, we obtain the following theorem.

**Theorem 3** *Let $\mathbf{w}^1, \mathbf{w}^2, \ldots$ be the sequence generated by DAL algorithm (Equations 19 and 20) and let $W^*$ be the set of minimizers of the objective* (1)*. Let us assume that there are a positive constant $\sigma$ and a scalar $\alpha$ ($1 \leq \alpha \leq 2$) such that*

**(A1)** $$f(\mathbf{w}^{t+1}) - f(W^*) \geq \sigma\|\mathbf{w}^{t+1} - W^*\|^\alpha \qquad (t = 0, 1, 2, \ldots), \tag{41}$$

*where $f(W^*)$ denotes the minimum objective value, and $\|\mathbf{w} - W^*\|$ denotes the minimum distance between $\mathbf{w} \in \mathbb{R}^n$ and the set of minimizers $W^*$ as $\|\mathbf{w} - W^*\| := \min_{\mathbf{w}^* \in W^*} \|\mathbf{w} - \mathbf{w}^*\|$.*

*If the inner minimization is solved exactly, we have the following inequality:*

$$\|\mathbf{w}^{t+1} - W^*\| + \sigma\eta_t \|\mathbf{w}^{t+1} - W^*\|^{\alpha - 1} \leq \|\mathbf{w}^t - W^*\|.$$

*Moreover, this implies that*

$$\|\mathbf{w}^{t+1} - W^*\|^{\frac{1+(\alpha-1)\sigma\eta_t}{1+\sigma\eta_t}} \leq \frac{1}{1+\sigma\eta_t}\|\mathbf{w}^t - W^*\|. \tag{42}$$

*That is, $\mathbf{w}^t$ converges to $W^*$* super-linearly *if $\alpha < 2$ or $\alpha = 2$ and $\eta_t$ is increasing, in a* global and non-asymptotic *sense.*

**Proof** See Appendix C.1. ■

Note that the above super-linear convergence holds without the assumption in Theorem 2 that $\eta_t$ is increased exponentially.

### 5.2 Approximate Inner Minimization

First we present a finite tolerance version of Lemma 1.

**Lemma 4** *Let $\mathbf{w}^1, \mathbf{w}^2, \ldots$ be the sequence generated by DAL algorithm (Equations 19 and 20). Let us assume the following conditions.*

**(A2)** *The loss function $f_\ell$ has a Lipschitz continuous gradient with modulus $1/\gamma$, that is,*

$$\left\|\nabla f_\ell(\mathbf{z}) - \nabla f_\ell(\mathbf{z}')\right\| \leq \frac{1}{\gamma}\|\mathbf{z} - \mathbf{z}'\| \qquad (\forall \mathbf{z}, \mathbf{z}' \in \mathbb{R}^m), \tag{43}$$

**(A3)** *The proximation with respect to $\phi_\lambda$ (see Equation 15) can be computed exactly.*

**(A4)** *The inner minimization (Equation 20) is solved to the following tolerance:*

$$\|\nabla \varphi_t(\alpha^t)\| \leq \sqrt{\frac{\gamma}{\eta_t}}\|\mathbf{w}^{t+1} - \mathbf{w}^t\|, \tag{44}$$

*where $\gamma$ is the constant in Equation* (43)*.*

*Under assumptions* **(A2)–(A4)**, *for arbitrary $\mathbf{w} \in \mathbb{R}^n$ we have*

$$\eta_t(f(\mathbf{w}^{t+1}) - f(\mathbf{w})) \leq \frac{1}{2}\|\mathbf{w}^t - \mathbf{w}\|^2 - \frac{1}{2}\|\mathbf{w}^{t+1} - \mathbf{w}\|^2. \tag{45}$$

**Proof** See Appendix C.2. ■

Note that Lemma 4 states that even with the weaker stopping criterion **(A4)**, we can obtain inequality (45) as in Lemma 1.

The assumptions we make here are rather weak. In Assumption **(A2)**, the loss function $f_\ell$ does not include the design matrix $\mathbf{A}$ (see Table 1). Therefore, it is easy to compute the constant $\gamma$. Accordingly, the stopping criterion **(A4)** can be checked without assuming anything about the data.

Furthermore, summing both sides of inequality (45) and assuming that $\eta_t$ is increased exponentially, we obtain Theorem 2 also under the approximate minimization **(A4)**.

Finally, an analogue of Theorem 3, which does not assume the exponential increase in $\eta_t$, is obtained as follows.

**Theorem 5** *Let $\mathbf{w}^1, \mathbf{w}^2, \ldots$ be the sequence generated by DAL algorithm and let $W^*$ be the set of minimizers of the objective* (1)*. Under assumption* **(A1)** *in Theorem 3 and* **(A2)-(A4)** *in Lemma 4, we have*

$$\|\mathbf{w}^{t+1} - W^*\|^2 + 2\sigma\eta_t\|\mathbf{w}^{t+1} - W^*\|^\alpha \leq \|\mathbf{w}^t - W^*\|^2,$$

*where $\|\mathbf{w}^t - W^*\|$ is the minimum distance between $\mathbf{w}^t$ and $W^*$ as in Theorem 3. Moreover, this implies that*

$$\|\mathbf{w}^{t+1} - W^*\|^{\frac{1+\alpha\sigma\eta_t}{1+2\sigma\eta_t}} \leq \frac{1}{\sqrt{1+2\sigma\eta_t}}\|\mathbf{w}^t - W^*\|. \tag{46}$$

*That is, $\mathbf{w}^t$ converges to $W^*$ super-linearly if $\alpha < 2$ or $\alpha = 2$ and $\eta_t$ is increasing.*

**Proof** Let $\bar{\mathbf{w}}^t$ be the closest point in $W^*$ from $\mathbf{w}^t$, that is, $\bar{\mathbf{w}}^t := \mathrm{argmin}_{\mathbf{w}^* \in W^*} \|\mathbf{w}^t - \mathbf{w}^*\|$. Using Lemma 4 with $\mathbf{w} = \bar{\mathbf{w}}^t$ and Assumption (**A1**), we have the first part of the theorem as follows:

$$\begin{aligned}\|\mathbf{w}^t - W^*\|^2 = \|\mathbf{w}^t - \bar{\mathbf{w}}^t\|^2 &\geq \|\mathbf{w}^{t+1} - \bar{\mathbf{w}}^t\|^2 + 2\sigma\eta_t \|\mathbf{w}^{t+1} - W^*\|^\alpha \\ &\geq \|\mathbf{w}^{t+1} - W^*\|^2 + 2\sigma\eta_t \|\mathbf{w}^{t+1} - W^*\|^\alpha,\end{aligned}$$

where we used the minimality of $\|\mathbf{w}^{t+1} - \bar{\mathbf{w}}^{t+1}\|$ in the second line. The last part of the theorem (46) can be obtained in a similar manner as that of Theorem 3 using Young's inequality (see Appendix C.1). ■

### 5.3 A Faster Rate

The factor $1/\sqrt{1+2\sigma\eta_t}$ obtained under the approximate minimization **(A4)** (see inequality (46) in Theorem 5) is larger than that obtained under the exact inner minimization (see inequality (42) in Theorem 3); that is, the statement in Theorem 5 is weaker than that in Theorem 3.

Here we show that a better rate can also be obtained for approximate minimization if we perform the inner minimization to $O(\|\mathbf{w}^{t+1} - \mathbf{w}^t\|/\eta_t)$ instead of $O(\|\mathbf{w}^{t+1} - \mathbf{w}^t\|/\sqrt{\eta_t})$ in Assumption **(A4)**.

**Theorem 6** *Let $\mathbf{w}^1, \mathbf{w}^2, \ldots$ be the sequence generated by DAL algorithm and let $W^*$ be the set of minimizers of the objective* (1)*. Under assumption* **(A1)** *in Theorem 3 with $\alpha = 2$, and assumptions* **(A2)** *and* **(A3)** *in Lemma 4, for any $\varepsilon < 1$ such that $\delta := (1-\varepsilon)/(\sigma\eta_t) \leq 3/4$, if we solve the inner minimization to the following precision*

$$\textbf{(A4}') \qquad \|\nabla\varphi_t(\alpha^t)\| \leq \frac{\sqrt{\gamma(1-\varepsilon)/\sigma}}{\eta_t}\|\mathbf{w}^{t+1} - \mathbf{w}^t\|,$$

*then we have*

$$\|\mathbf{w}^{t+1} - W^*\| \leq \frac{1}{1+\varepsilon\sigma\eta_t}\|\mathbf{w}^t - W^*\|.$$

**Proof** See Appendix C.3 ■

Note that the assumption $\delta < 3/4$ is rather weak, because if the factor $\delta$ is greater than one, the stopping criterion (**A4**′) would be weaker than the earlier criterion (**A4**). In order to be on the safe side, we can choose $\varepsilon = \max(\varepsilon_0, 1 - 3\sigma\eta_t/4)$ (assuming that we know the constant $\sigma$) and the above statement holds with $\varepsilon = \varepsilon_0$. Unfortunately, in exchange for obtaining a faster rate, the stopping criterion **(A4′)** now depends not only on $\gamma$, which can be computed, but also on $\sigma$, which is hard to know in practice. Therefore stopping condition **(A4′)** is not practical.

### 5.4 Validity of Assumption (A1)

In this subsection, we discuss the validity of assumption **(A1)** and its relation to the assumptions used in Rockafellar (1976a) and Kort and Bertsekas (1976). Roughly speaking, our assumption **(A1)** is milder than the one used in Rockafellar (1976a) and stronger than the one used in Kort and Bertsekas (1976).

First of all, assumption **(A1)** is unnecessary for convergence in terms of function value (Theorem 2 and its approximate version implied by Lemma 4). Exponential increase of the proximity

parameter $\eta_t$ may sound restrictive, but this is the setting we typically use in experiments (see Section 7.1). Assumption **(A1)** is only necessary in Theorem 3 and Theorem 5 to translate the residual function value $f(\mathbf{w}^{t+1}) - f(W^*)$ into the residual distance $\|\mathbf{w}^{t+1} - W^*\|$.

We can roughly think of Assumption **(A1)** as a *local strong convexity* assumption. Here we say a function $f$ is *locally strongly convex around* the set of minimizers $W^*$ if for all positive $C$, all the points $\mathbf{w}$ within distance $C$ from the set $W^*$, the objective function $f$ is bounded below by a quadratic function, that is,

$$f(\mathbf{w}) - f(W^*) \geq \sigma \|\mathbf{w} - W^*\|^2 \quad (\forall \mathbf{w} : \|\mathbf{w} - W^*\| \leq C), \tag{47}$$

where the positive constant $\sigma$ may depend on $C$. If the set of minimizers $W^*$ is bounded, all the level sets of $f$ are bounded (see Rockafellar, 1970, Theorem 8.7). Therefore, if we make sure that the function value $f(\mathbf{w}^t)$ does not increase during the minimization, we can assume that all points generated by DAL algorithm are contained in some neighborhood around $W^*$ that contains the level set defined by the initial function value $\{\mathbf{w} \in \mathbb{R}^n : f(\mathbf{w}) \leq f(\mathbf{w}^0)\}$; that is, the local strong convexity of $f$ guarantees Assumption **(A1)** with $\alpha = 2$.

Note that Kort and Bertsekas (1976, p278) used a slightly weaker assumption than the local strong convexity (47); they assumed that there *exists* a positive constant $C' > 0$ such that the local strong convexity (47) is true for all $\mathbf{w}$ in the neighborhood $\|\mathbf{w} - W^*\| \leq C'$ for some $\sigma > 0$.

The local strong convexity (47) or Assumption **(A1)** fails when the objective function behaves like a constant function around the set of minimizers $W^*$. In this case, DAL converges rapidly in terms of function value due to Theorem 2; however it does not necessarily converge in terms of the distance $\|\mathbf{w}^t - W^*\|$.

Note that the objective function $f$ is the sum of the loss term and the regularization term. Even if the minimum eigenvalue of the Hessian of the loss term is very close to zero, we can hope that the regularization term holds the function up from the minimum objective value $f(W^*)$. For example, when the loss term is *zero* and we only have the $\ell_1$-regularization term $\phi_\lambda^{\ell_1}(\mathbf{w})$. The objective $f(\mathbf{w}) = \lambda \sum_{j=1}^n |w_j|$ can be lower-bounded as

$$f(\mathbf{w}) \geq \frac{\lambda}{C} \|\mathbf{w}\|^2 \quad (\forall \mathbf{w} : \|\mathbf{w}\| \leq C),$$

where the minimizer $\mathbf{w}^*$ is $\mathbf{w}^* = \mathbf{0}$. Note that the $\ell_1$-regularizer is not (globally) strongly convex. The same observation holds also for other regularizers we discussed in Section 4.

In the context of asymptotic analysis of AL algorithm, Rockafellar (1976b) assumed that there exists $\tau > 0$, such that in the ball $\|\beta\| \leq \tau$ in $\mathbb{R}^n$, the gradient of the convex conjugate $f^*$ of the objective function $f$ is Lipschitz continuous with constant $L$, that is,

$$\|\nabla f^*(\beta) - \nabla f^*(0)\| \leq L \|\beta\|.$$

Note that because $\partial f(\nabla f^*(0)) \ni 0$ (Rockafellar, 1970, Corollary 23.5.1), $\nabla f^*(0)$ is the optimal solution $\mathbf{w}^*$ of Equation (1), and it is unique by the continuity assumed above.

Our assumption **(A1)** can be justified from Rockafellar's assumption as follows.

**Theorem 7** *Rockafellar's assumption implies that the objective $f$ is locally strongly convex with $C = c\tau L$ and $\sigma = \min(1, (2c-1)/c^2)/(2L)$ for any positive constant $c$ ($\tau$ and $L$ are constants from Rockafellar's assumption).*

**Proof** The proof is a *local* version of the proof of Theorem X.4.2.2 in Hiriart-Urruty and Lemaréchal (1993) (Lipschitz continuity of $\nabla f^*$ implies strong convexity of $f$). See Appendix C.4. ■

Note that as the constant $c$ that bounds the distance to the set of minimizers $W^*$ increases, the constant $\sigma$ becomes smaller and the convergence guarantee in Theorem 3 and 5 become weaker (but still valid).

Nevertheless Assumption **(A1)** we use in Theorem 3 and 5 are weaker than the local strong convexity (47), because we need Assumption **(A1)** to hold only on the points generated by DAL algorithm. For example, if we only consider a finite number of steps, such a constant $\sigma$ always exists.

Both assumptions in Rockafellar (1976b) and Kort and Bertsekas (1976) are made for asymptotic analysis. In fact, they require that as the optimization proceeds, the solution becomes closer to the optimum $\mathbf{w}^*$ in the sense of the distance $\|\mathbf{w}^t - W^*\|$ in Kort and Bertsekas (1976) and $\|\beta\|$ in Rockafellar (1976b). However in both cases, it is hard to predict how many iterations it takes for the solution to be sufficiently close to the optimum so that the super-linear convergence happens.

Our analysis is complementary to the above classical results. We have shown that super-linear convergence happens *non-asymptotically* under Assumption **(A1)**, which is trivial for a finite number of steps. Assumption **(A1)** can also be guaranteed for infinite steps using the local strong convexity around $W^*$ (47).

## 6. Previous Studies

In this section, we discuss earlier studies in two categories. The first category comprises methods that try to overcome the difficulty posed by the nondifferentiability of the regularization term $\phi_\lambda(\mathbf{w})$. The second category, which includes DAL algorithm in this paper, consists of methods that try to overcome the difficulty posed by the coupling (or non-separability) introduced by the design matrix $\mathbf{A}$. The advantages and disadvantages of all the methods are summarized in Table 3.

### 6.1 Constrained Optimization, Upper-Bound Minimization, and Subgradient Methods

Many authors have focused on the *nondifferentiability* of the regularization term in order to efficiently minimize Equation (1). This view has lead to three types of approaches, namely, (i) constrained optimization, (ii) upper-bound minimization, and (iii) subgradient methods.

In the constrained optimization approach, auxiliary variables are introduced to rewrite the nondifferentiable regularization term as a linear function of conically-constrained auxiliary variables. For example, the $\ell_1$-norm of a vector $\mathbf{w}$ can be rewritten as:

$$\|\mathbf{w}\|_1 = \sum_{j=1}^{n} \min_{w_j^{(+)}, w_j^{(-)} \geq 0} \left( w_j^{(+)} + w_j^{(-)} \right) \quad \text{s.t.} \quad w_j = w_j^{(+)} - w_j^{(-)},$$

where $w_j^{(+)}$ and $w_j^{(-)}$ $(j = 1, \ldots, n)$ are auxiliary variables and they are constrained in the positive-orthant cone. Two major challenges of the auxiliary-variable formulation are the increased size of the problem and the complexity of solving a constrained optimization problem.

The projected gradient (PG) method (see Bertsekas, 1999) iteratively computes a gradient step and projects it back to the constraint-set. The PG method in Figueiredo et al. (2007b) converges

R-linearly,[7] if the loss function is quadratic. However, PG methods can be extremely slow when the design matrix $\mathbf{A}$ is poorly conditioned. To overcome the scaling problem, the L-BFGS-B algorithm (Byrd et al., 1995) can be applied for the simple positive orthant constraint that arises from the $\ell_1$ minimization. However, this approach does not easily extend to more general regularizers, such as group lasso and trace norm regularization.

The interior-point (IP) method (see Boyd and Vandenberghe, 2004) is another algorithm that is often used for constrained minimization; see Koh et al. 2007; Kim et al. 2007 for the application of IP methods to sparse estimation problems. Basically an IP method generates a sequence that approximately follows the so called central path, which parametrically connects the analytic center of the constraint-set and the optimal solution. Although IP methods can tolerate poorly conditioned design matrices well, it is challenging to scale them up to very large dense problems. The convergence of the IP method in Koh et al. (2007) is empirically found to be linear.

The second approach (upper-bound minimization) constructs a differentiable upper-bound of the nondifferentiable regularization term. For example, the $\ell_1$-norm of a vector $\mathbf{w}$ can be rewritten as follows:

$$\|\mathbf{w}\|_1 = \sum_{j=1}^{n} \min_{\alpha_j \geq 0} \left( \frac{w_j^2}{2\alpha_j} + \frac{\alpha_j}{2} \right). \tag{48}$$

In fact, the right-hand side is an upper bound of the left-hand side for arbitrary non-negative $\alpha_j$ due to the inequality of arithmetic and geometric means, and the equality is obtained by setting $\alpha_j = |w_j|$. The advantage of the above parametric-upper-bound formulation is that for a fixed set of $\alpha_j$, the problem (2) becomes a (weighted) quadratically regularized minimization problem, for which various efficient algorithms already exist. The iteratively reweighted shrinkage (IRS) method (Gorodnitsky and Rao, 1997; Bioucas-Dias, 2006; Figueiredo et al., 2007a) alternately solves the quadratically regularized minimization problem and tightens (re-weights) the upper-bound in Equation (48). A more general technique was studied in parallel by the name of variational EM (Jaakkola, 1997; Girolami, 2001; Palmer et al., 2006), which generalizes the above upper-bound using Fenchel's inequality (Rockafellar, 1970). A similar approach that is based on Jensen's inequality (Rockafellar, 1970) has been studied in the context of multiple-kernel learning (Micchelli and Pontil, 2005; Rakotomamonjy et al., 2008) and in the context of multi-task learning (Argyriou et al., 2007, 2008). The challenge in the IRS framework is the *singularity* (Figueiredo et al., 2007a) around the coordinate axis. For example, in the $\ell_1$-problem in Equation (2), any zero component $w_j = 0$ in the initial vector $\mathbf{w}$ will remain zero after any number of iterations. Moreover, it is possible to create a situation that the convergence becomes arbitrarily slow for finite $|w_j|$ because the convergence in the $\ell_1$ case is only linear (Gorodnitsky and Rao, 1997).

The third approach (subgradient methods) directly handles the nondifferentiability through subgradients; see, for example, Bertsekas (1999).

A (stochastic) subgradient method typically converges as $O(1/\sqrt{k})$ for non-smooth problems in general and as $O(1/k)$ if the objective is strongly convex; see Shalev-Shwartz et al. (2007); Lan (2010). However, since the method is based on gradients, it can easily fail when the problem is poorly conditioned (see, e.g., Yu et al., 2010, Section 2.2). Therefore, one of the challenges in subgradient-based approaches is to take the second-order curvature information into account. This

7. A sequence $\xi^t$ converges to $\xi$ R-linearly (R is for "root") if the residual $|\xi^t - \xi|$ is bounded by a sequence $\varepsilon^t$ that linearly converges to zero (Nocedal and Wright, 1999).

| | Constrained Optimization | | Upper-bound Minimization | Subgradient Method | Iterative Proximation | |
|---|---|---|---|---|---|---|
| | PG | IP | IRS | OWLQN | AG | DAL |
| Poorly conditioned **A** | – | ✓ | ✓ | ✓ | – | ✓ |
| No singularity | ✓ | ✓ | – | ✓ | ✓ | ✓ |
| Extensibility | ✓ | ✓ | ✓ | – | ✓ | ✓ |
| Exploits sparsity of **w** | ✓ | – | – | ✓ | ✓ | ✓ |
| Efficient when | – | – | – | $m \gg n$ | $m \gg n$ | $m \ll n$ |
| Convergence | $(O(e^{-k}))$ | $(O(e^{-k}))$ | $O(e^{-k})$ | ? | $O(1/k^2)$ | $o(e^{-k})$ |

Table 3: Comparison of the algorithms to solve Equation (1). In the columns, six methods, namely, projected gradient (PG), interior point (IP), iterative reweighted shrinkage (IRS), orthant-wise limited-memory quasi Newton (OWLQN), accelerated gradient (AG), and dual augmented Lagrangian (DAL), are categorized into four groups discussed in the text. The first row: "Poorly conditioned **A**" means that a method can tolerate poorly conditioned design matrices well. The second row: "No singularity" means that a method does not suffer from singularity in the parametrization (see main text). The third row: "Extensibility" means that a method can be easily extended beyond $\ell_1$-regularization. The forth row: "Exploits sparsity of **w**" means that a method can exploit the sparsity in the intermediate solution. The fifth row: "Efficient when" indicates the situations each algorithm runs efficiently, namely, more samples than unknowns ($m \gg n$), more unknowns than samples ($m \ll n$), or does not matter (–). The last row shows the rate of convergence known from literature. The super-linear convergence of DAL is established in this paper.

is especially important to tackle large-scale problems with a possibly poorly conditioned design matrix. Orthant-wise limited memory quasi Newton (OWLQN, Andrew and Gao, 2007) and sub-LBFGS (Yu et al., 2010) combine subgradients with the well known L-BFGS quasi Newton method (Nocedal and Wright, 1999). Although being very efficient for $\ell_1$-regularization and piecewise linear loss functions, these methods depend on the efficiency of oracles that compute a descent direction and a step-size; therefore, it is challenging to extend these methods to combinations of general loss functions and general nondifferentiable regularizers. In addition, the convergence rates of the OWLQN and subLBFGS methods are not known.

### 6.2 Iterative Proximation

Yet another approach is to deal with the nondifferentiable regularization through the proximity operation. In fact, the proximity operator (15) is easy to compute for many practically relevant separable regularizers.

The remaining issue, therefore, is the coupling between variables introduced by the design matrix **A**. We have shown in Sections 3.2 and 3.3 that IST and DAL can be considered as two different strategies to remove this coupling.

Recently many studies have focused on methods that iteratively compute the proximal operation (15) (Figueiredo and Nowak, 2003; Daubechies et al., 2004; Combettes and Wajs, 2005; Nesterov, 2007; Beck and Teboulle, 2009; Cai et al., 2008), which can be described in an abstract manner as

follows:

$$\mathbf{w}^{t+1} = \text{prox}_{\phi_{\lambda_t}}\left(\mathbf{y}^t\right), \tag{49}$$

where $\text{prox}_{\phi_\lambda}$ is the proximal operator defined in Equation (15). The above mentioned studies can be differentiated by the different $\mathbf{y}^t$ and $\lambda_t$ that they use.

For example, the IST approach (also known as the *forward-backward splitting* Lions and Mercier, 1979; Combettes and Wajs, 2005; Duchi and Singer, 2009) can be described as follows:

$$\begin{aligned}\mathbf{y}^t &:= \mathbf{w}^t - \eta_t \mathbf{A}^\top \nabla f_\ell(\mathbf{A}\mathbf{w}^t),\\ \lambda_t &:= \lambda\eta_t.\end{aligned}$$

What we need to do at every iteration is only to compute the gradient at the current point, take a gradient step, and then perform the proximal operation (Equation 49). Note that $\eta_t$ can be considered as a step-size.

The IST method can be considered as a generalization of the projected gradient method. Since the proximal gradient step (13) reduces to an ordinary gradient step when $\phi_\lambda = 0$, the basic idea behind IST is to keep the non-smooth term $\phi_\lambda$ as a part of the proximity step (see Lan, 2010). Consequently, the convergence behaviour of IST is the same as that of (projected) gradient descent on the differentiable loss term. Note that Duchi and Singer (2009) analyze the case where the loss term is also nondifferentiable in both batch and online learning settings. Langford et al. (2009) also analyze the online setting with a more general threshold operation.

IST approach maintains sparsity of $\mathbf{w}^t$ throughout the optimization, which results in significant reduction of computational cost; this is an advantage of iterative proximation methods compared to interior-point methods (e.g., Koh et al., 2007), because the solution produced by interior-point methods becomes sparse only in an asymptotic sense; see Boyd and Vandenberghe (2004).

The downside of the IST approach is the difficulty to choose the step-size parameter $\eta_t$; this issue is especially problematic when the design matrix $\mathbf{A}$ is poorly conditioned. In addition, the best known convergence rate of a naive IST approach is $O(1/k)$ (Beck and Teboulle, 2009), which means that the number of iterations $k$ that we need to obtain a solution $\mathbf{w}^k$ such that $f(\mathbf{w}^k) - f(\mathbf{w}^*) \leq \varepsilon$ grows linearly with $1/\varepsilon$, where $f(\mathbf{w}^*)$ is the minimal value of Equation (1).

SpaRSA (Wright et al., 2009) uses approximate second order curvature information for the selection of the step-size parameter $\eta_t$. TwIST (Bioucas-Dias and Figueiredo, 2007) is a "two-step" approach that tries to alleviate the poor efficiency of IST when the design matrix is poorly conditioned. However the convergence rates of SpaRSA and TwIST are unknown.

Accelerating strategies that use different choices of $\mathbf{y}^t$ have been proposed in Nesterov (2007) and Beck and Teboulle (2009) (denoted AG in Tab. 3), which have $O(1/k^2)$ guarantee with almost the same computational cost per iteration; see also Lan (2010).

DAL can be considered as a new member of the family of iterative proximation algorithms. We have qualitatively shown in Section 3.3 that DAL constructs a better lower bound of the loss term than IST. Moreover, we have rigorously studied the convergence rate of DAL and have shown that it converges super-linearly. Of course the fast convergence of DAL comes with the increased cost per iteration. Nevertheless, as we have qualitatively discussed in Section 4, this increase is mild, because the sparsity of intermediate solutions can be effectively exploited in the inner minimization. We empirically compare DAL with other methods in Section 7.

There is of course an issue on how much one should precisely optimize when the training error (plus the regularization term) is a crude approximation of the generalization error (Shalev-Shwartz

and Srebro, 2008). However the reason we use sparse regularization is exactly that we are not only interested in the predictive power. We argue that when we are using sparse methods to gain insights into some problem, it is important that we are sure that we are doing what we write in our paper (e.g., "solve an $\ell_1$-regularized minimization problem"), and someone else can reliably recover the same sparsity pattern using any optimization approach that employs some objective stopping criterion such as the duality gap. Of course the stability of the optimal solution itself must be analyzed (see Bickel et al., 2009; Zhao and Yu, 2006; Meinshausen and Bühlmann, 2006) and the trade-off between accuracy and sparsity should be discussed. However, this is beyond the scope of this paper.

## 7. Empirical Results

In this section, we confirm the super-linear convergence of DAL algorithm and compare it with other algorithms on $\ell_1$-regularized logistic regression problems. The algorithms that we compare are FISTA (Beck and Teboulle, 2009), OWLQN (Andrew and Gao, 2007), SpaRSA (Wright et al., 2009), IRS (Figueiredo et al., 2007a), and L1_LOGREG (Koh et al., 2007). Note that IST is not included because SpaRSA and FISTA are shown to clearly outperform the naive IST approach. We describe the logistic regression problem and the implementation of all of the methods in Section 7.1. The synthetic experiments are presented in Section 7.2 and the benchmark experiments are presented in Section 7.3.

### 7.1 Implementation

In this subsection, we first describe the problem to be solved and then explain the implementation of the above mentioned algorithms in detail.

For all algorithms except for IRS, the initial solution $\mathbf{w}^0$ was set to an all zero vector. For IRS, the initial solution was sampled from an independent standard Gaussian distribution.

The CPU time was measured on a Linux server with two 3.1 GHz Opteron Processors and 32GB of RAM.

#### 7.1.1 $\ell_1$-REGULARIZED LOGISTIC REGRESSION

The logistic regression model is defined by the loss function

$$f_{\mathrm{LR}}(\mathbf{z}) = \sum_{i=1}^{m} \ell_{\mathrm{LR}}(z_i, y_i) := \sum_{i=1}^{m} \log(1 + e^{-y_i z_i}), \tag{50}$$

where $y_i \in \{-1, +1\}$ is a training label. The conjugate of the loss function can be obtained as follows:

$$f^*_{\mathrm{LR}}(-\alpha) = \sum_{i=1}^{m} \ell^*_{\mathrm{LR}}(-\alpha_i, y_i),$$

where

$$\ell^*_{\mathrm{LR}}(-\alpha_i, y_i) = \begin{cases} \alpha_i y_i \log(\alpha_i y_i) + (1 - \alpha_i y_i)\log(1 - \alpha_i y_i) & (\text{if } 0 \le \alpha_i y_i \le 1), \\ +\infty & (\text{otherwise}). \end{cases}$$

Rewriting the dual problem (3) we have the following expression:

$$\underset{\alpha\in\mathbb{R}^m}{\text{maximize}} \qquad -f^*_{\mathrm{LR}}(-\alpha), \tag{51}$$

$$\text{subject to} \qquad \|\mathbf{A}^\top\alpha\|_\infty \le \lambda, \tag{52}$$

where $\|\mathbf{y}\|_\infty = \max_{j=1,\ldots,n}|y_j|$ is the $\ell_\infty$-norm; note that the implicit constraint in Equation (3) (through the indicator function $\delta^\infty_\lambda$) is made explicit in Equation (52).

For the experiments in this section, we reparametrize the regularization constant $\lambda$ as $\lambda = \bar{\lambda}\|\mathbf{A}^\top\mathbf{y}\|_\infty$. The reason for this reparametrization is that for all $\bar{\lambda} \ge 0.5$ the solution $\mathbf{w}$ can be shown to be zero; thus we can measure the strength of the regularization relative to the problem using $\bar{\lambda}$ instead of $\lambda$. This is because the conjugate loss function $f^*_{\mathrm{LR}}$ takes the minimum at $\alpha_i = y_i/2$ and the minimum is attained for $\lambda \ge \|\mathbf{A}^\top(\mathbf{y}/2)\|_\infty$ (see Equation 52).

### 7.1.2 DUALITY GAP

We used the relative duality gap (RDG) as a stopping criterion with tolerance $10^{-3}$. More specifically, we terminated all the algorithms described below when RDG fell below $10^{-3}$. RDG was computed as follows for all algorithms except L1_LOGREG. For L1_LOGREG, we modified the stopping criterion implemented in the original code by the authors from absolute duality gap to relative duality gap. See also Koh et al. (2007), Wright et al. (2009) and Tomioka and Sugiyama (2009).

Let $\bar{\alpha}^t$ be any candidate dual vector at $t$th iteration. For example, $\bar{\alpha}^t = \alpha^t$ for DAL and $\bar{\alpha}^t = -\nabla f_\ell(\mathbf{A}\mathbf{w}^{t+1})$ for OWLQN, SpaRSA, and IRS. Note that the above $\bar{\alpha}^t$ does not necessarily satisfy the dual constraint (52). Thus we define $\tilde{\alpha}^t = \bar{\alpha}^t \min(1, \lambda/\|\mathbf{A}^\top\bar{\alpha}^t\|_\infty)$. Notice that $\|\mathbf{A}^\top\tilde{\alpha}^t\|_\infty \le \lambda$ by construction. We compute the dual objective value as $d(\mathbf{w}^{t+1}) = -f^*_\ell(-\tilde{\alpha}^t)$; see Equation (51). Finally $\mathrm{RDG}^{t+1}$ is obtained as $\mathrm{RDG}^{t+1} = (f(\mathbf{w}^{t+1}) - d(\mathbf{w}^{t+1}))/f(\mathbf{w}^{t+1})$, where $f$ is the primal objective function defined in Equation (1).

The norm of the minimum norm subgradient is also frequently used as a stopping criterion. However, there are two reasons for using RDG instead. First, the gradient at the current point is not evaluated in FISTA (Beck and Teboulle, 2009) and it requires additional computation, whereas the vector $\alpha^t$ in the computation of RDG does not need to be the gradient at the current point; in fact the gradient at any point (or any $m$-dimensional vector) gives a valid lower bound of the minimum objective value. Second, since the gradient can change discontinuously at nondifferentiable points, the norm of gradient does not reflect the distance from the solution well; this is a problem for, for example, an interior-point method, because it produces a sparse solution only asymptotically.

### 7.1.3 DAL

DAL algorithm is implemented in MATLAB.[8] The inner minimization problem (see Equation 10) is solved with Newton's method; we used the preconditioned conjugate gradient (PCG) method for solving the associated Newton system (`pcg` function in MATLAB); we use the diagonal elements of the Hessian matrix (see Equation 25) as the preconditioner. The inner minimization is terminated by the criterion (44) with $\gamma = 4$, because the Hessian of the loss function (50) is uniformly bounded by $1/4$ (see Table 1).

8. The software is available from `http://www.ibis.t.u-tokyo.ac.jp/ryotat/dal/`.

We chose the initial proximity parameter to be either $\eta_0 = 0.01/\lambda$ (conservative setting) or $\eta_0 = 1/\lambda$ (aggressive setting) and increased $\eta_t$ by a factor of 2 at every iteration. Since $\eta_t$ appears in the soft-thresholding operation multiplied by $\lambda$, it seems to be intuitive to choose $\eta_t$ inversely proportional to $\lambda$ but we do not have a formal argument yet. We empirically discuss the choice of $\eta_0$ in more detail in Section 7.2.4.

The algorithm was terminated when the RDG fell below $10^{-3}$.

### 7.1.4 DAL-B

DAL-B is a variant of DAL with an unregularized bias term (see update Equations 35-37). This algorithm is included because L1_LOGREG implemented by Koh et al. (2007) estimates a bias term and therefore cannot be directly compared to DAL.

As an augmented Lagrangian algorithm, DAL-B solves the following dual problem:

$$\begin{aligned} \underset{\alpha\in\mathbb{R}^m}{\text{maximize}} \quad & -f^*_{\mathrm{LR}}(-\alpha) - \delta^\infty_\lambda(\mathbf{v}), \\ \text{subject to} \quad & \mathbf{A}^\top\alpha = \mathbf{v}, \qquad (53) \\ & \mathbf{1}^\top\alpha = 0. \qquad (54) \end{aligned}$$

See also Equations (3) and (4).

When implementing DAL-B, we noticed that sometimes the algorithm gets stuck in a plateau where the additional equality constraint (54) improves very little. This was more likely to happen when the condition of the design matrix was poor.

In order to avoid this undesirable slow-down, we heuristically adapt the proximity parameter $\eta_t$ for the equality constraint (54). Note that this kind of modification cannot improve the theoretical convergence result without additional prior information. More specifically, we use proximity parameters $\eta_t^{(1)}$ and $\eta_t^{(2)}$ for equality constraints (53) and (54), respectively. The AL function (37) is rewritten as follows

$$\alpha^t = \underset{\alpha\in\mathbb{R}^m}{\operatorname{argmin}} \left( f^*_\ell(-\alpha) + \frac{1}{2\eta_t^{(1)}} \|\operatorname{prox}^{\ell_1}_{\lambda\eta_t}(\mathbf{w}^t + \eta_t^{(1)}\mathbf{A}^\top\alpha)\|^2 + \frac{1}{2\eta_t^{(2)}}(b^t + \eta_t^{(2)}\mathbf{1}_m{}^\top\alpha)^2 \right).$$

First we initialize $\eta_0^{(1)} = \eta_0^{(2)} = 0.01/\lambda$ (conservative setting) or $\eta_0^{(1)} = \eta_0^{(2)} = 1/\lambda$ (aggressive setting) as above. The proximity parameter $\eta_t^{(1)}$ with respect to Equation (53) is increased by the factor 2 at every iteration (the same as DAL). The proximity parameter $\eta_t^{(2)}$ with respect to Equation (54) is increased by a larger factor 40 if the following conditions are satisfied:

1. The iteration counter $t > 1$.
2. The violation of the equality constraint (54), namely $\mathrm{viol}^t := |\mathbf{1}^\top\alpha^t|$, does not sufficiently decrease; that is, $\mathrm{viol}^t > \mathrm{viol}^{t-1}/2$.
3. The violation $\mathrm{viol}^t$ is larger than $10^{-3}$ (the tolerance of optimization).

Otherwise, $\eta_t^{(2)}$ is increased by the same factor 2 as $\eta_t^{(1)}$.

Note that the theoretical results in Section 5 still holds if we replace $\eta_t$ in Section 5 by $\eta_t^{(1)}$, because $\eta_t^{(1)} \leq \eta_t^{(2)}$; that is, the stopping criterion (44) and the convergence rates simply become more conservative.

Since DAL-B has an additional equality constraint (54). We modified the computation of relative duality gap described above by defining the candidate dual vector $\bar{\alpha}^t$ as $\bar{\alpha}^t = \alpha^t - \frac{1}{m}\mathbf{1}_m\mathbf{1}_m{}^\top\alpha^t$.

#### 7.1.5 FAST ITERATIVE SHRINKAGE-THRESHOLDING ALGORITHM (FISTA)

FISTA algorithm (Beck and Teboulle, 2009) is implemented in MATLAB. The algorithm is terminated by the same RDG criterion except that the dual objective is evaluated at $\mathbf{y}^t$ in update equation (49) instead of $\mathbf{w}^{t+1}$; this approach saves unnecessary computation of gradients.

#### 7.1.6 ORTHANT-WISE LIMITED MEMORY QUASI NEWTON (OWLQN)

OWLQN algorithm (Andrew and Gao, 2007) is also implemented in MATLAB because we found that our MATLAB implementation was faster than the C++ implementation provided by the authors; this is because MATLAB uses optimized linear algebra routines while authors' implementation does not. The algorithm is terminated by the same RDG criterion as DAL.

#### 7.1.7 SPARSE RECONSTRUCTION BY SEPARABLE APPROXIMATION (SPARSA)

SpaRSA algorithm (Wright et al., 2009) is implemented in MATLAB. We modified the code provided by the authors[9] to handle the logistic loss function. The algorithm is terminated by the same RDG criterion.

#### 7.1.8 ITERATIVE REWEIGHTED SHRINKAGE (IRS)

IRS algorithm is implemented in MATLAB. At every iteration IRS solves a ridge-regularized logistic regression problem with the regularizer defined in Equation (48). This problem can be converted into a standard $\ell_2$-regularized logistic regression with the design matrix $\tilde{\mathbf{A}} = \mathbf{A}\mathrm{diag}(\sqrt{\alpha_1}, \ldots, \sqrt{\alpha_n})$ by reparametrizing $w_j$ to $\tilde{w}_j = w_j/\sqrt{\alpha_j}$. The weight $\alpha_j$ is set to $|w_j^t|$ before solving the problem. Thus if any $w_j^t = 0$, the corresponding column of $\tilde{\mathbf{A}}$ becomes zero and it can be removed from the optimization. We use the limited memory BFGS quasi-Newton method (Nocedal and Wright, 1999) to solve each sub-problem.

#### 7.1.9 INTERIOR POINT ALGORITHM (L1_LOGREG)

L1_LOGREG algorithm (Koh et al., 2007) is implemented in C. We modified the code provided by the authors[10] as a C-MEX function so that it can be called directly from MATLAB without saving matrices into files. We used the BLAS and LAPACK libraries provided together with MATLAB R2008b (`-lmwblas` and `-lmwlapack` options for the `mex` command). L1_LOGREG is also terminated by the RDG criterion.

Note that L1_LOGREG also estimates an unregularized bias term. DAL algorithm with a bias term (DAL-B) is included to make the comparison easy; see Section 7.1.4.

9. Code can be found at `http://www.lx.it.pt/~mtf/SpaRSA/`.
10. Code can be found at `http://www.stanford.edu/~boyd/l1_logreg/`.

### 7.2 Synthetic Experiment

In this subsection, we first confirm the convergence behaviour of DAL (Section 7.2.2); second we compare the scaling of various algorithms against the size of the problem (Section 7.2.3); finally we discuss how to choose the initial proximity parameter $\eta_0$ (Section 7.2.4).

#### 7.2.1 EXPERIMENTAL SETTING

The elements of the design matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ were randomly sampled from an independent standard Gaussian distribution. The true classifier coefficient $\beta$ was generated by filling randomly chosen element (4%) of a $n$-dimensional vector with samples from an independent standard Gaussian distribution; the remaining elements of the vector were set to zero. The training label vector $\mathbf{y}$ was obtained by taking the sign of $\mathbf{A}\beta + 0.01\xi$, where $\xi \in \mathbb{R}^m$ was a sample from an $m$-dimensional independent standard Gaussian distribution. The whole procedure was repeated ten times.

#### 7.2.2 EMPIRICAL VALIDATION OF SUPER-LINEAR CONVERGENCE

In this section, we empirically confirm the validity of the convergence results (Theorems 2, 3 and 5) obtained in the previous section and compare the efficiency of DAL, FISTA, OWLQN, SpaRSA, and IRS for the number of samples $m = 1,024$ and the number of parameters $n = 16,384$. L1_LOGREG is not included because it solves a different minimization problem. We use the regularization constant $\bar{\lambda} = 0.01$. For DAL, we used the aggressive setting ($\eta_t = 1/\lambda, 2/\lambda, 4/\lambda, \ldots$).

First in order to obtain the true minimizer[11] $\mathbf{w}^*$ of Equation (1), we ran DAL algorithm to obtain a solution with high precision (RDG $< 10^{-9}$). Assuming that the support of this solution is correct, we performed one Newton step of Equation (1) in the subspace of active variables. The solution $\mathbf{w}^*$ we obtained in this way satisfied $\|\nabla f(\mathbf{w}^*)\| < 10^{-13}$, where $\nabla f(\mathbf{w}^*)$ is the minimum norm subgradient of $f$ at $\mathbf{w}^*$. The parameter $\sigma$ in Equation (41) was estimated by taking the minimum of $(f(\mathbf{w}^t) - f(\mathbf{w}^*))/\|\mathbf{w}^t - \mathbf{w}^*\|^2$ along the trajectory obtained by the above minimization and multiplying the minimum value by a safety factor of 0.7. In order to estimate the residual norm $\|\mathbf{w}^t - \mathbf{w}^*\|$, we use bounds (42) and (46) with $\alpha = 2$ and the initial residual $\|\mathbf{w}^0 - \mathbf{w}^*\|$. The bound (40) in Theorem 2 is used with the same initial residual to estimate the reduction in the function value.

In Figure 2, we show a result of a typical (single) run of the algorithms described above. Note that the result is not averaged to keep the meaning of theoretical bounds.

In the top left panel of Figure 2, we can see that the convergence in terms of the norm of the residual vector $\mathbf{w}^t - \mathbf{w}^*$ happens indeed rapidly as predicted by the theorems in Section 5. The yellow curve shows the result of Theorem 3, which assumes exact minimization of Equation (20), and the magenta curve shows the result of Theorem 5, which allows some error in the minimization of Equation (20). We can see that the difference between the optimistic analysis of Theorem 3 and the realistic analysis of Theorem 5 is negligible. In this problem, in order to reach the quality of solution DAL achieves in 10 iterations OWLQN and SpaRSA take at least 100 iterations and FISTA takes 1,000 iterations. The IRS approach required about the same number of iterations as OWLQN and SpaRSA but each step was much heavier than those two algorithms (see also the top right panel in Figure 2) and it was terminated after 100 iterations.

11. We assume that the minimizer is unique.

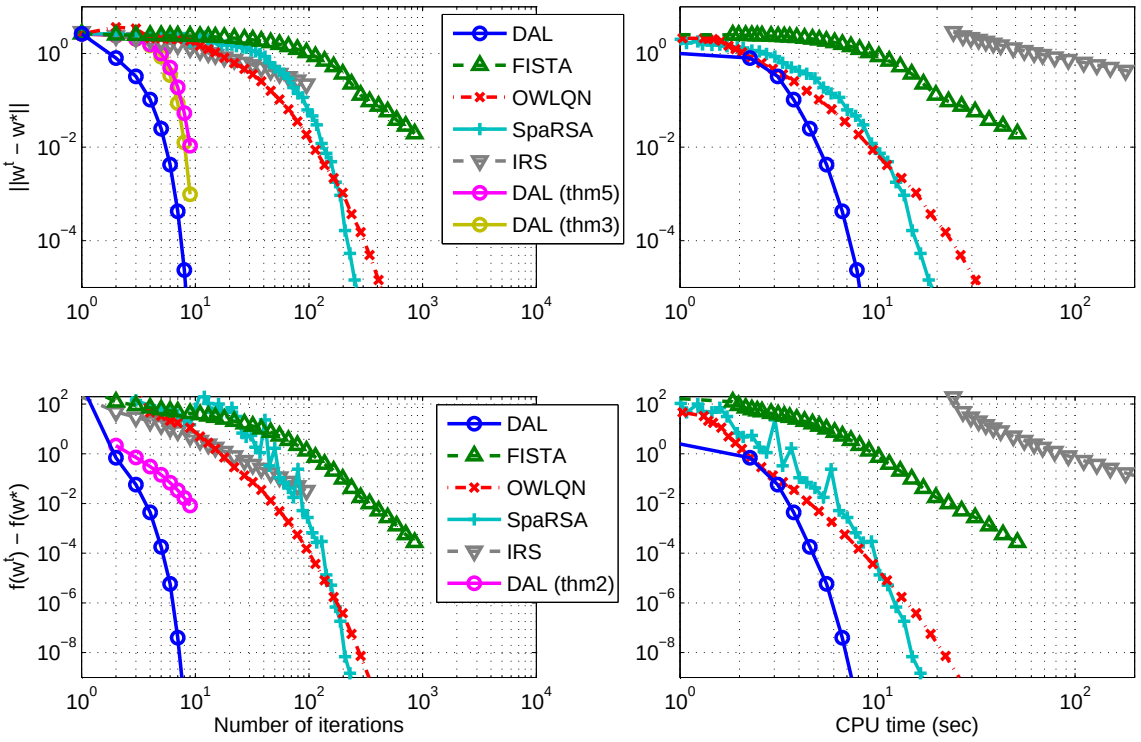



Figure 2: Empirical comparison of DAL, FISTA (Beck and Teboulle, 2009), OWLQN (Andrew and Gao, 2007), and SpaRSA (Wright et al., 2009). Top left: residual norm vs. number of iterations. Also the theoretical guarantees for DAL from Theorems 3 and 5 are shown. Top right: residual norm vs. CPU time. Bottom left: residual in the function value vs. number of iterations. Bottom right: residual in the function value vs. CPU time.

The bottom left panel of Figure 2 shows comparison of five algorithms DAL, FISTA, OWLQN, SpaRSA, and IRS in terms of the decrease in the function value. Also plotted is the decrease in the function value predicted by Theorem 2 (magenta curve). The convergence of DAL is the fastest also in terms of function value. OWLQN and SpaRSA are the next after DAL and are faster than FISTA.

DAL needs to solve a minimization problem at every iteration. Accordingly the operation required in each iteration is heavier than those in FISTA, OWLQN, and SpaRSA. Thus we compare the total CPU time spent by the algorithms in the right part of Figure 2. It can be seen that DAL can obtain a solution that is much more accurate in less than 10 seconds than the solution FISTA obtained after almost 60 seconds. In terms of computation time, DAL and OWLQN seem to be on par at low precision. However as the precision becomes higher DAL becomes clearly faster than OWLQN. SpaRSA seems to be slightly slower than DAL and OWLQN.

Two algorithms (DAL-B and L1_LOGREG) that also estimate an unregularized bias term are compared in Figure 3. The number of observations $m = 1,024$ and the number of parameters $n = 16,384$, and all other settings are identical as above. A variant of DAL-B that does not use the heuristics described in Section 7.1.4 is included for comparison. For DAL-B without the heuristics,

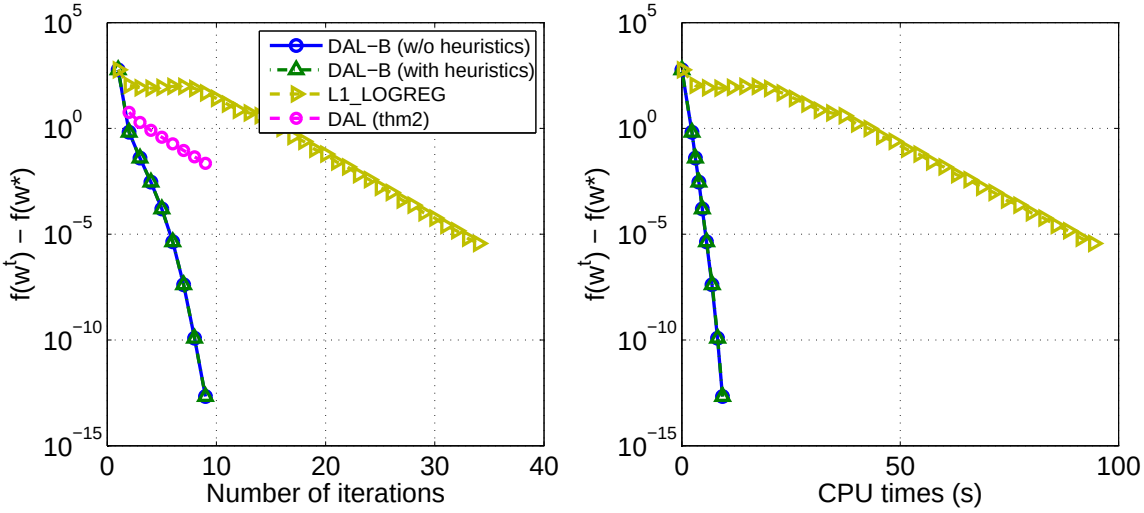



Figure 3: Comparison of DAL-B and L1_LOGREG (Koh et al., 2007). Both algorithms estimate an unregularized bias term. The left panel shows the residual function value against the number of iterations. The right panel shows the same against the CPU time spent by the algorithms.

the proximity parameters $\eta_t^{(1)}$ and $\eta_t^{(2)}$ are both initialized to $1/\lambda$ and increased by the factor 2. For DAL-B with the heuristics, the proximity parameter $\eta_t^{(2)}$ is increased more aggressively; see Section 7.1.4.

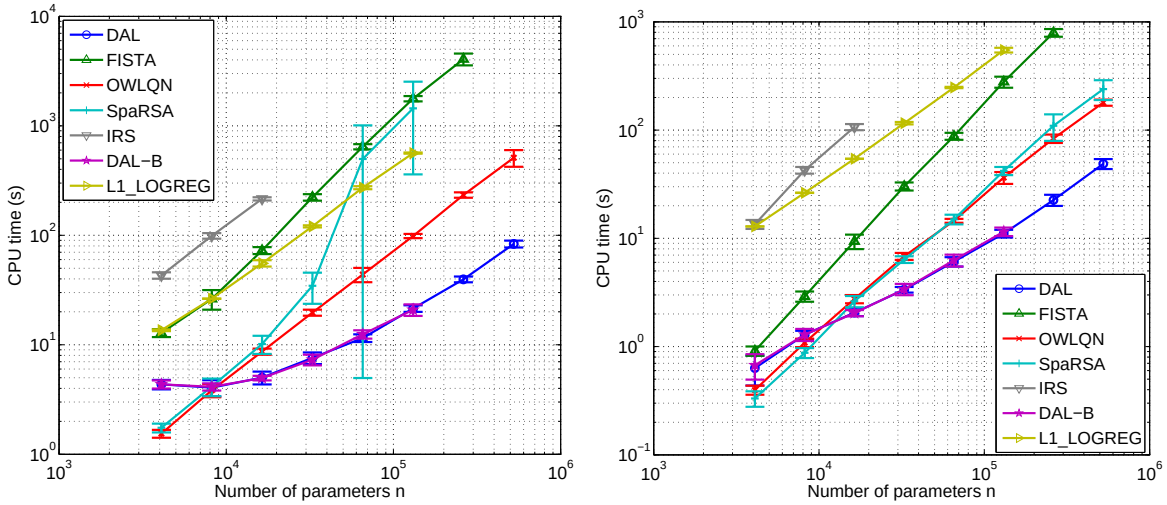
In the left panel in Figure 3, the residual of primal objective values of both algorithms are plotted against the number of iterations. As empirically observed in Koh et al. (2007), L1_LOGREG converges linearly; after roughly 10 iterations, the residual function value reduces by a factor around 2 in each iteration (a factor 1.85 was reported in Koh et al., 2007). The convergence of DAL-B is faster than L1_LOGREG and the curve is slightly concave downwards, which indicates the super-linearity of the convergence. Note also that the linear convergence bound from Theorem 2 is shown. The heuristics described in Section 7.1.4 shows almost no effect on this problem, probably because the design matrix is well conditioned.

The right panel in Figure 3 shows the same information against the CPU time spent by the algorithms. DAL-B is roughly 10 times faster than L1_LOGREG to achieve residual less than $10^{-5}$.

### 7.2.3 SCALING AGAINST THE SIZE OF THE PROBLEM

Here we compare how well different algorithms scale against the number of parameters $n$. We fixed the number of samples $m$ at $m = 1,024$ and varied the number of parameters from $n = 4,096$ to $n = 524,288$. We used two regularization constants $\bar{\lambda} = 0.1$ and $\bar{\lambda} = 0.01$.

The results are summarized in Figure 4. Figures 4(a) and 4(b) show the results for $\bar{\lambda} = 0.01$ and $\bar{\lambda} = 0.1$, respectively. In each figure we plot the CPU time spent to reach RDG$< 10^{-3}$ against the number of parameters $n$.




(a) $\bar{\lambda} = 0.01$. (b) $\bar{\lambda} = 0.1$.

Figure 4: CPU time of various algorithms on synthetic logistic regression problems.

One can see that DAL has the mildest dependence on the number of parameters among the methods compared. In particular, DAL is faster than other algorithms for roughly $n > 10^4$. Also note that DAL and DAL-B show similar scaling against the number of parameters; that is, adding an unregularized bias term has no significant influence on the computational efficiency.

For $\bar{\lambda} = 0.01$, SpaRSA shows sharp increase in the CPU time from around $n = 32{,}768$, which is similar to the result in Tomioka and Sugiyama (2009) (Figure 3). Also notice the increased error-bar. In fact, for $n \geq 65{,}536$, it had to be stopped after 5,000 iterations in some runs, whereas it converged after few hundred iterations in other runs. On the other hand, SpaRSA scales similarly to OWLQN and is more stable for $\bar{\lambda} = 0.1$.

For all algorithms except L1_LOGREG, solving the problem for larger regularization constant $\bar{\lambda} = 0.1$ requires less computation than for $\bar{\lambda} = 0.01$. Nevertheless the advantage of the DAL algorithm is larger for the more computationally demanding situation of $\bar{\lambda} = 0.01$ against FISTA, OWLQN, SpaRSA, and IRS. On the other hand, the advantage of DAL against L1_LOGREG is larger for $\bar{\lambda} = 0.1$, because the CPU time of L1_LOGREG is almost constant in both cases. The CPU time of DAL with (DAL-B) and without (DAL) the bias term are almost the same.

### 7.2.4 CHOICE OF $\eta_0$

In this subsection, we show how the choice of the sequence $\eta_t$ changes the behaviour of DAL algorithm. We ran DAL algorithm for $\bar{\lambda} = 0.1$ with $\eta_0 = 1/\lambda$ (as above), which we call the aggressive setting, and $\eta_0 = 0.01/\lambda$, which we call the conservative setting. In both cases, $\eta_t$ is increased by a factor of 2 as in the previous experiments. No bias term is used.

In Figure 5, plotted are the number of PCG steps for inner minimization and the CPU time spent by DAL algorithm with the conservative setting ($\eta_0 = 0.01/\lambda$, left) and the aggressive setting ($\eta_0 = 1/\lambda$, right). The average number of PCG steps and CPU time are shown as stacked bar-plots, in which each segment of a bar corresponds to one outer iteration. One can see that in the

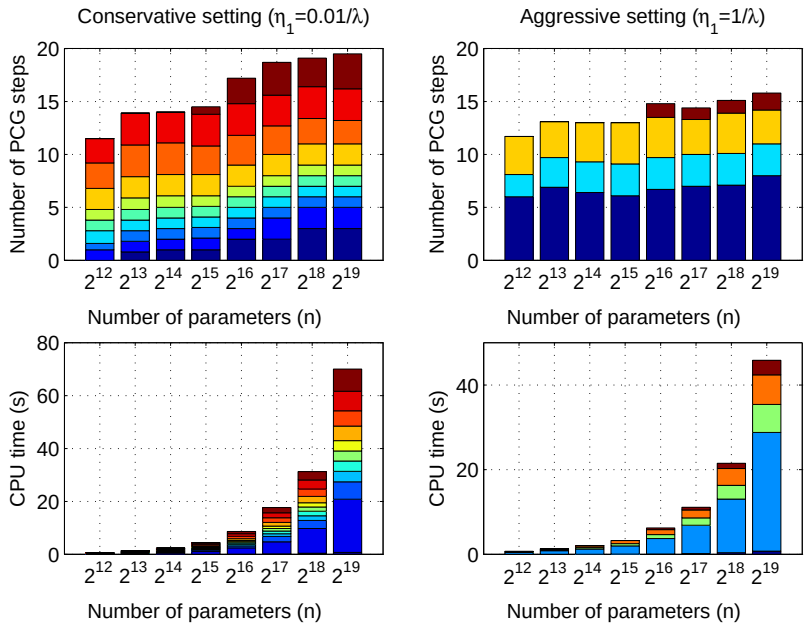



Figure 5: Comparison of behaviours of DAL algorithm for different choices of initial proximity parameter $\eta_0$. Left: $\eta_0 = 0.01/\lambda$ (conservative setting). Right: $\eta_0 = 1/\lambda$ (aggressive setting). On the top row, the cumulative numbers of PCG steps (inner steps) are shown. On the bottom row, the cumulative CPU time spent by the algorithm is shown.

conservative setting, DAL uses roughly 8 to 10 outer iterations, whereas in the aggressive setting, the number of outer iterations is reduced to less than a half (3 or 4). On the other hand, the total number of PCG steps is only slightly smaller in the aggressive setting. Therefore, in the aggressive setting DAL spends more PCG steps for each outer iteration. It is worth noting that almost half of the PCG iterations are spent for the first outer iteration in the aggressive setting, whereas in the conservative setting the PCG steps are more distributed. In terms of the CPU time, the aggressive setting is about 10–30% faster than the conservative setting because it saves both computation required for each outer-iteration and inner-iteration. However, generally speaking increasing the proximity parameter $\eta_t$ makes the condition of the problem worse; in fact we found that the algorithm did not always converge for $\eta_0 = 100/\lambda$. Thus it is not recommended to use too large value for $\eta_t$.

Figure 6 compares the total CPU time spent by the two variants of DAL for $\overline{\lambda} = 0.1$. As discussed above, the aggressive setting ($\eta_0 = 1/\lambda$) is faster than the conservative setting ($\eta_0 = 0.01/\lambda$). However the difference is minor compared to the change in the proximity parameter $\eta_0$,

### 7.3 Benchmark Data Sets

In this subsection, we apply the algorithms discussed in the previous subsection except IRS to benchmark data sets, and compare their efficiency on various problems; IRS is omitted because it was clearly outperformed by other methods on the synthetic data.

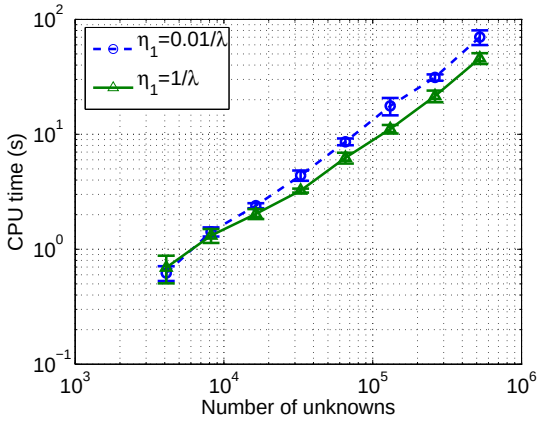



Figure 6: Comparison of conservative ($\eta_0 = 0.01/\lambda$) and aggressive ($\eta_0 = 1/\lambda$) choice of proximity parameter $\eta_0$ for $\bar{\lambda} = 0.1$ (see also Figure 4(b)). Note that the aggressive setting is used in Sections 7.2.2 and 7.2.3 and the conservative setting is used in Section 7.3.

### 7.3.1 EXPERIMENTAL SETTING

The benchmark data sets we use are five data sets from NIPS 2003 Feature Selection Challenge,[12] 20 newsgroups data set,[13] and a bioinformatics data[14] provided by Baranzini et al. (2004)

The five data sets from the Feature Selection Challenge (**arcene, dexter**, **dorothea**, **gisette**, and **madelon**) are all split into training-, validation-, and test-set. We combine the training- and validation-sets and randomly split each data set into a training-set that contains two-thirds of the examples, and a test-set that contains the remaining one-third. We apply the $\ell_1$-regularized logistic regression solvers to the training-set and report the accuracy on the test-set as well as the CPU time for training. This procedure was repeated 10 times (also for the two other data sets below). The numbers of training instances and features, and the format of each data set (sparse or dense) are summarized in Table 4.

From the 20 newsgroups data set (**20news**), we deal with the binary classification of category "alt.atheism" vs. "comp.graphics". We use the preprocessed MATLAB format data. The original data set consists of $1,061$ training examples and 707 test examples. We again combine all the examples and randomly split them into a training-set containing two-thirds of the examples and a test-set containing the rest. The training example has $n = 61,188$ features which are provided as a sparse matrix.

The goal in Baranzini et al. (2004) is to predict the response (good or poor) to recombinant human interferon beta (rIFNβ) treatment of multiple sclerosis patients from gene-expression measurements. The data set is denoted as **gene**. The data set consists of gene-expression profile of 70 genes from 52 subjects. We again randomly select two-thirds of the subjects for training and the

12. The data sets are available from `http://www.nipsfsc.ecs.soton.ac.uk/datasets/`; see Guyon et al. (2006) for more information.
13. The data set is available from `http://people.csail.mit.edu/jrennie/20Newsgroups/`.
14. The data is available from `http://www.plosbiology.org/article/info:doi/10.1371/journal.pbio.0030002`.

remaining for testing. Following the setting in the original paper, we used only the expression data from the beginning of the treatment ($t = 0$) and preprocessed the data by taking all the polynomials up to third order, that is, we compute (i) $x$, $x^2$, and $x^3$ for each single feature $x$, (ii) $xy$, $x^2y$, and $xy^2$ for every pair of features $(x,y)$, and (iii) $xyz$ for every triplet of features $(x,y,z)$. As a result we obtain 62,195 $(= 3 \cdot 70 + 3 \cdot 2{,}415 + 54{,}740)$ features.

In every data set, we standardized each feature to zero mean and unit standard deviation before applying the algorithms. Since the standardized design matrix $\tilde{\mathbf{A}}$ is usually dense even if the original matrix $\mathbf{A}$ is sparse, we provide function handles that compute $\tilde{\mathbf{A}}\mathbf{x}$ and $\tilde{\mathbf{A}}^\top\mathbf{y}$ instead of $\tilde{\mathbf{A}}$ itself with DAL, FISTA, OWLQN, and SpaRSA. This can be done by keeping the vector of means and standard deviations of the original design matrix as follows:

$$\tilde{\mathbf{A}}\mathbf{x} = \mathbf{A}\mathbf{S}^{-1}\mathbf{x} - \mathbf{1}_m\mathbf{m}^\top\mathbf{S}^{-1}\mathbf{x},$$
$$\tilde{\mathbf{A}}^\top\mathbf{y} = \mathbf{S}^{-1}\mathbf{A}^\top(\mathbf{y} - \frac{1}{m}\mathbf{1}_m\mathbf{1}_m{}^\top\mathbf{y}),$$

where $\mathbf{m} \in \mathbb{R}^n$ is the vector of means and $\mathbf{S}$ is a $n \times n$ diagonal matrix that has the standard deviations of the original features on the diagonal. If the standard deviation of any feature is zero, we placed one in the corresponding element of $\mathbf{S}$. L1_LOGREG is implemented with a similar technique; see Koh et al. (2007).

We compare the CPU time that is necessary to compute the whole regularization path. In order to define the regularization path, we choose 20 log-linearly separated values from $\bar{\lambda} = 0.5$ to $\underline{\lambda} = 0.001$, where $\lambda$ is the normalized regularization constant defined in Section 7.1.1. We apply a warm start strategy to all the algorithms; that is, we sequentially solve problems for smaller and smaller regularization constants using the solution obtained from the last optimization (for a larger regularization constant) as the initial solution.

All the methods were terminated when the relative duality gap fell below $10^{-3}$. For DAL algorithms (DAL and DAL-B) we choose the conservative setting, that is, we initialize $\eta_0^{(1)}$ and $\eta_0^{(2)}$ as $0.01/\lambda$.

### 7.3.2 RESULTS

Table 4 summarizes the problems and the performance of the algorithms. For each algorithm, we show the maximum test accuracy obtained in the regularization path and the CPU time spent to compute the whole path. The smallest and the second smallest CPU times are shown in bold-face and italic, respectively. One can see that DAL is the fastest in most cases when the number of parameters $n$ is larger than the number of observations. In addition, the CPU time of two variants of DAL (with and without the bias term) tend to be similar except **dorothea** data set. For most data sets, the accuracy obtained by DAL algorithm is close to FISTA, OWLQN, and SpaRSA, and the accuracy obtained by DAL-B is close to L1_LOGREG.

Figure 7 illustrates a typical situation where DAL algorithm is efficient. Since the size of the inner minimization problem (20) is proportional to the number of observations $m$, when $n \gg m$, DAL is more efficient than other methods that work in the primal.

In contrast, Figure 8 illustrates the situation where DAL is not very efficient compared to other algorithms. In Figure 8, we can also see that for all algorithms except L1_LOGREG, the cost of solving one minimization problem grows larger as the regularization constant is reduced, whereas the cost seems almost constant for L1_LOGREG.

| | | arcene | dexter | dorothea | gisette | madelon | 20news | gene |
|---|---|---|---|---|---|---|---|---|
| **m** | | 133 | 400 | 767 | 4667 | 1733 | 1179 | 35 |
| **n** | | 10000 | 20000 | 100000 | 5000 | 500 | 61188 | 62195 |
| **format** | | dense | sparse | sparse | dense | dense | sparse | dense |
| **DAL** | accuracy | 70.60 | 91.75 | 93.71 | 97.70 | 61.53 | 92.84 | 82.35 |
| | time (s) | **3.47** | *4.20* | 36.61 | *77.02* | 16.73 | *28.10* | *5.56* |
| **DAL-B** | accuracy | 72.54 | 92.00 | 93.05 | 97.71 | 61.43 | 92.87 | 81.18 |
| | time (s) | *3.56* | *4.77* | **10.60** | *82.96* | 17.96 | *26.31* | **5.49** |
| **FISTA** | accuracy | 70.60 | 91.75 | 93.79 | 97.71 | 61.51 | 92.80 | 82.35 |
| | time (s) | 25.34 | 7.24 | 284.59 | **52.19** | *10.40* | *27.95* | 108.27 |
| **OWLQN** | accuracy | 70.60 | 91.75 | 93.76 | 97.70 | 61.56 | 92.82 | 82.35 |
| | time (s) | 17.63 | *5.25* | 134.31 | *70.96* | 19.08 | *23.11* | 132.21 |
| **SpaRSA** | accuracy | 70.90 | 91.75 | 93.71 | 97.70 | 61.55 | 95.14 | 78.24 |
| | time (s) | 294.80 | 29.98 | 1377.20 | *91.65* | *10.11* | 310.96 | 1622.26 |
| **L1_LOG-REG** | accuracy | 72.84 | 92.05 | 93.05 | 97.71 | 61.48 | 92.85 | 81.18 |
| | time (s) | 8.98 | **3.39** | 109.92 | *98.37* | **5.90** | **21.48** | 16.58 |

Table 4: Results on benchmark data sets. We tested six algorithms, namely, DAL, DAL-B, FISTA, OWLQN, SpaRSA, L1_LOGREG on seven benchmark data sets. See main text for the description of the data sets. $m$ is the number of observations. $n$ is the number of features. For each algorithm, shown are the test accuracy and the CPU time spent to compute the regularization path with a warm-start strategy. All the numbers are averaged over 10 runs. Bold face numbers indicate the fastest CPU time. Italic numbers indicate CPU times that are within two times of the fastest CPU time.

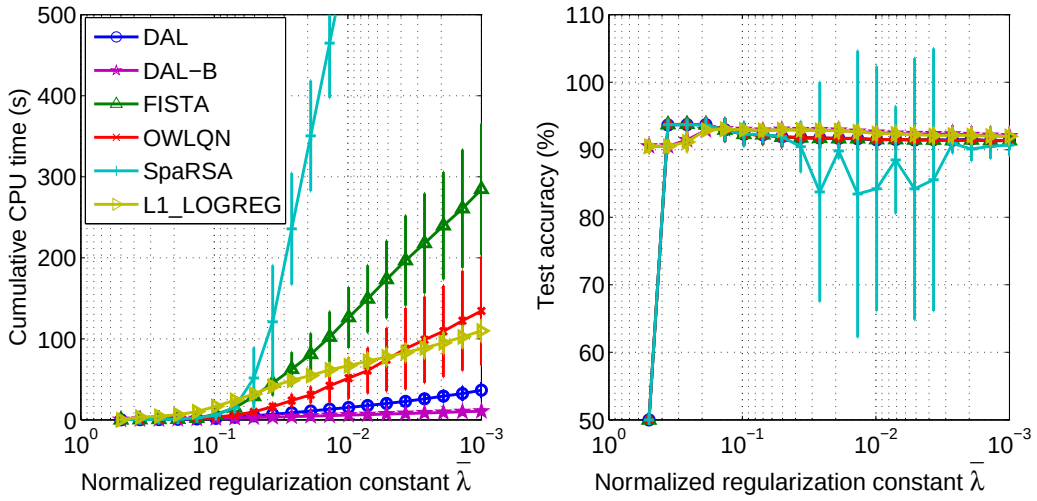



Figure 7: **Dorothea** data set ($m = 767$, $n = 100{,}000$). DAL is efficient in this case ($m \ll n$).

## 8. Conclusion

In this paper, we have extended DAL algorithm (Tomioka and Sugiyama, 2009) for general regularized minimization problems, and provided it with a new view based on the proximal minimization

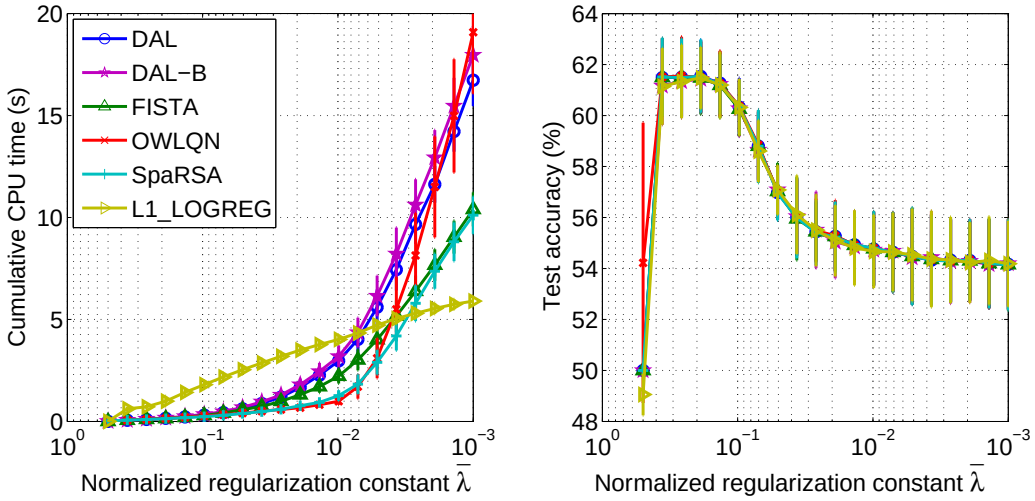



Figure 8: **Madelon** data set ($m = 1{,}733$, $n = 500$). DAL is not very efficient in this case ($m \gg n$).

framework in Rockafellar (1976b). Generalizing the recent result from Beck and Teboulle (2009), we improved the convergence results on super-linear convergence of augmented Lagrangian methods in literature for the case of sparse estimation.

Importantly, most assumptions that we made in our analysis can be checked independent of data. Instead of assuming that the problem is strongly convex we assume that the loss function has a Lipschitz continuous gradient, which can be checked before receiving data. Another assumption we have made is that the proximation with respect to the regularizer can be computed analytically, which can also be checked without looking at data. Moreover, we have shown that such assumption is valid for the $\ell_1$-regularizer, group lasso regularizer, and any other support function of some convex set for which the projection onto the set can be analytically obtained.

Compared to the general result in Rockafellar (1976b), our result is stronger when the inner minimization is solved approximately. Compared to Kort and Bertsekas (1976), we do not need to assume the strong convexity of the objective function, which is obviously violated for the dual of many sparsity regularized estimation problems; instead we assume that the loss function has Lipschitz continuous gradient. Note that we use no asymptotic arguments as in Rockafellar (1976b) and Kort and Bertsekas (1976). Currently, our results does not apply to primal-based augmented Lagrangian method discussed in Goldstein and Osher (2009) for loss functions that are not strongly convex (e.g., logistic loss). The extension of our analysis to these methods is a future work.

The theoretically predicted rapid convergence of DAL algorithm is also empirically confirmed in simulated $\ell_1$-regularized logistic regression problems. Moreover, we have compared six recently proposed algorithms for $\ell_1$-regularized logistic regression, namely DAL, FISTA, OWLQN, SpaRSA, L1_LOGREG, and IRS on synthetic and benchmark data sets. On the synthetic data sets, we have shown that DAL has the mildest dependence on the number of parameters among the methods compared. On the benchmark data sets, we have shown that DAL is the fastest among the methods compared when the number of parameters is larger than the number of observations on both sparse and dense data sets.

Furthermore, we have empirically investigated the relationship between the choice of the initial proximity parameter $\eta_0$ and the number of (inner/outer) iterations as well as the computation time. We found that the computation can be sped up by choosing a large value for $\eta_0$; however the improvement is often small compared to the change in $\eta_0$ and choosing large value for $\eta_0$ can make the inner minimization unstable by making the problem poorly conditioned.

There are basically two strategies to make an efficient optimization algorithm. One is to use many iterations that are very light. FISTA, SpaRSA, and OWLQN (and also stochastic approaches Shalev-Shwartz and Srebro, 2008; Duchi and Singer, 2009) fall into this category. Theoretical convergence guarantee is often weak for these methods, for example, $O(1/k^2)$ for FISTA. Another strategy is to use a small number of heavier iterations. Interior point methods, such as L1_LOGREG, are prominent examples of this class. DAL can be considered as a member of the second class. We have theoretically and empirically shown that DAL requires a small number of outer iterations. At the same time, DAL inherits good properties of iterative shrinkage/thresholding algorithms from the first class. For example, it effectively uses the fact that the proximal operation can be computed analytically, and it can maintain the sparsity of the parameter vector during optimization. Furthermore, we have shown that the dual formulation of DAL makes the inner minimization efficient, because (i) typically the number of observations $m$ is smaller than the number of parameters $n$, and (ii) the gradient and Hessian of the inner objective can be computed efficiently for sparse estimation problems.

Future work includes the extension of our analysis to the primal-based augmented Lagrangian methods (Yin et al., 2008; Goldstein and Osher, 2009; Yang and Zhang, 2009; Lin et al., 2009), application of approximate augmented Lagrangian methods and operator splitting methods to machine learning problems (see Zhang et al., 2010; Boyd et al., 2011; Tomioka et al., 2011b), and application of DAL to more advanced sparse estimation problems (e.g., Cai et al., 2008; Wipf and Nagarajan, 2008; Tomioka et al., 2011a).

## Acknowledgments

We would like to thank Masakazu Kojima, Masao Fukushima, and Hisashi Kashima for helpful discussions. This work was partially supported by the Global COE program "The Research and Training Center for New Development in Mathematics", MEXT KAKENHI 22700138, 22700289, and the FIRST program.

## Appendix A. Preliminaries on Proximal Operation

This section contains some basic results on proximal operation, which we use in later sections and is based on Moreau (1965), Rockafellar (1970), and Combettes and Wajs (2005).

### A.1 Proximal Operation

Let $f$ be a closed proper convex function over $\mathbb{R}^n$ that takes values in $\mathbb{R}\cup\{+\infty\}$. The *proximal operator* with respect to $f$ is defined as follows:

$$\mathrm{prox}_f(\mathbf{z}) = \operatorname*{argmin}_{\mathbf{x}\in\mathbb{R}^n}\left(f(\mathbf{x})+\frac{1}{2}\|\mathbf{x}-\mathbf{z}\|^2\right). \tag{55}$$

Note that because of the strong convexity of the minimand in the right-hand side, the above minimizer is unique. Similarly we define the proximal operator with respect to the convex conjugate function $f^*$ of $f$ as follows:

$$\mathrm{prox}_{f^*}(\mathbf{z}) = \operatorname*{argmin}_{\mathbf{x}\in\mathbb{R}^n} \left( f^*(\mathbf{x}) + \frac{1}{2}\|\mathbf{x}-\mathbf{z}\|^2 \right).$$

The following elegant result is well known.

**Lemma 8 (Moreau's decomposition)** *The proximation of a vector $\mathbf{z}\in\mathbb{R}^n$ with respect to a convex function $f$ and that with respect to its convex conjugate $f^*$ is complementary in the following sense:*

$$\mathrm{prox}_f(\mathbf{z}) + \mathrm{prox}_{f^*}(\mathbf{z}) = \mathbf{z}.$$

**Proof** The proof can be found in Moreau (1965) and Rockafellar (1970, Theorem 31.5). Here, we present a slightly more simple proof.

Let $\mathbf{x} = \mathrm{prox}_f(\mathbf{z})$ and $\mathbf{y} = \mathrm{prox}_{f^*}(\mathbf{z})$. By definition we have $\partial f(\mathbf{x}) + \mathbf{x} - \mathbf{z} \ni 0$ and $\partial f^*(\mathbf{y}) + \mathbf{y} - \mathbf{z} \ni 0$, which imply

$$\partial f(\mathbf{x}) \ni \mathbf{z}-\mathbf{x}, \tag{56}$$
$$\partial f^*(\mathbf{z}-\mathbf{x}) \ni \mathbf{x}, \tag{57}$$

and

$$\partial f^*(\mathbf{y}) \ni \mathbf{z}-\mathbf{y}, \tag{58}$$
$$\partial f(\mathbf{z}-\mathbf{y}) \ni \mathbf{y}, \tag{59}$$

respectively, because $(\partial f)^{-1} = \partial f^*$ (Rockafellar, 1970, Corollary 23.5.1).

From Equations (56) and (58), we have

$$f(\mathbf{z}-\mathbf{y}) \geq f(\mathbf{x}) + (\mathbf{z}-\mathbf{y}-\mathbf{x})^\top(\mathbf{z}-\mathbf{x}), \tag{60}$$
$$f^*(\mathbf{z}-\mathbf{x}) \geq f^*(\mathbf{y}) + (\mathbf{z}-\mathbf{x}-\mathbf{y})^\top(\mathbf{z}-\mathbf{y}). \tag{61}$$

Similarly, Equations (57) and (59) give

$$f^*(\mathbf{y}) \geq f^*(\mathbf{z}-\mathbf{x}) + (\mathbf{y}-\mathbf{z}+\mathbf{x})^\top\mathbf{x}, \tag{62}$$
$$f(\mathbf{x}) \geq f(\mathbf{z}-\mathbf{y}) + (\mathbf{x}-\mathbf{z}+\mathbf{y})^\top\mathbf{y}. \tag{63}$$

Summing both sides of Equations (60)–(63), we have

$$0 \geq 2\|\mathbf{z}-\mathbf{x}-\mathbf{y}\|^2,$$

from which we conclude that $\mathbf{x}+\mathbf{y}=\mathbf{z}$. ■

Proximal operation can be considered as a generalization of the projection onto a convex set. For example, if we take $f$ as the indicator function of the $\ell_\infty$ ball of radius $\lambda$, that is, $f(\mathbf{z}) = \delta^\infty_\lambda(\mathbf{z})$ (see Equation 5), then the proximal operation with respect to $f$ is the projection onto the $\ell_\infty$-ball (8). On the other hand, the proximal operation with respect to the $\ell_1$-regularizer is the soft-thresholding operator (9). Therefore, we notice that

$$\mathrm{proj}_{[-\lambda,\lambda]}(\mathbf{z}) + \mathrm{prox}^{\ell_1}_\lambda(\mathbf{z}) = \mathbf{z},$$

which is a special case of Lemma 8, because the $\ell_1$-regularizer is the convex conjugate of the indicator function of the $\ell_\infty$-ball of radius $\lambda$; see Figure 9.

### A.2 Moreau's Envelope

The minimum attained in Equation (55) is called the Moreau envelope of $f$:

$$F(\mathbf{z}) = \min_{\mathbf{x}\in\mathbb{R}^n} \left( f(\mathbf{x}) + \frac{1}{2}\|\mathbf{x}-\mathbf{z}\|^2 \right). \tag{64}$$

The decomposition in Lemma 8 can be expressed in terms of envelope functions as follows.

**Lemma 9 (Decomposition and envelope functions)** *Let $f$ and $f^*$ be a pair of convex conjugate functions, and let $F$ and $F^*$ be the Moreau envelopes of $f$ and $f^*$, respectively. Then we have*

$$F(\mathbf{z}) + F^*(\mathbf{z}) = \frac{1}{2}\|\mathbf{z}\|^2.$$

**Proof** Let $\mathbf{x} = \text{prox}_f(\mathbf{z})$ and $\mathbf{y} = \text{prox}_{f^*}(\mathbf{z})$ as in the proof of Lemma 8. From the definition of convex conjugate $f^*$, we have

$$f(\mathbf{x}) + f^*(\mathbf{y}) = \mathbf{y}^\top \mathbf{x},$$

because $\mathbf{y} = \mathbf{z} - \mathbf{x} \in \partial f(\mathbf{x})$ (Rockafellar, 1970, Theorem 23.5). Therefore, we have

$$\begin{aligned} F(\mathbf{z}) + F^*(\mathbf{z}) &= f(\mathbf{x}) + \frac{1}{2}\|\mathbf{y}\|^2 + f^*(\mathbf{y}) + \frac{1}{2}\|\mathbf{x}\|^2 \\ &= \mathbf{y}^\top \mathbf{x} + \frac{1}{2}\|\mathbf{y}\|^2 + \frac{1}{2}\|\mathbf{x}\|^2 \\ &= \frac{1}{2}\|\mathbf{x}+\mathbf{y}\|^2 = \frac{1}{2}\|\mathbf{z}\|^2, \end{aligned}$$

where we used $\mathbf{x}+\mathbf{y} = \mathbf{z}$ from Lemma 8 in the last line. ■

Note that $F^*$ is the Moreau envelope of $f^*$ and *not* the convex conjugate of $F$.

Moreau's envelope can be considered as a inf-convolution (see Rockafellar, 1970) of $f$ and a quadratic function $\|\cdot\|^2/2$. Accordingly it is differentiable and the derivative is given in the following lemma.

**Lemma 10 (Derivative of Moreau's envelope)** *Moreau's envelope function $F$ in Equation* (64) *is continuously differentiable (even if $f$ is not differentiable) and the derivative can be written as follows:*

$$\nabla F(\mathbf{z}) = \text{prox}_{f^*}(\mathbf{z}).$$

**Proof** The proof can be found in Moreau (1965) and Rockafellar (1970, Theorem 31.5). We repeat the proof below for completeness. The proof consists of two parts. We first show that for all $\mathbf{z}, \mathbf{z}' \in \mathbb{R}^n$

$$F(\mathbf{z}') \geq F(\mathbf{z}) + (\mathbf{z}'-\mathbf{z})^\top \mathbf{y}, \tag{65}$$

where $\mathbf{y} = \text{prox}_{f^*}(\mathbf{z})$, which implies that $\mathbf{y} = \text{prox}_{f^*}(\mathbf{z}) \in \partial F(\mathbf{z})$. Second, we show that

$$F(\mathbf{z}') \leq F(\mathbf{z}) + (\mathbf{z}'-\mathbf{z})^\top \mathbf{y} + \frac{\|\mathbf{z}'-\mathbf{z}\|^2}{2}, \tag{66}$$

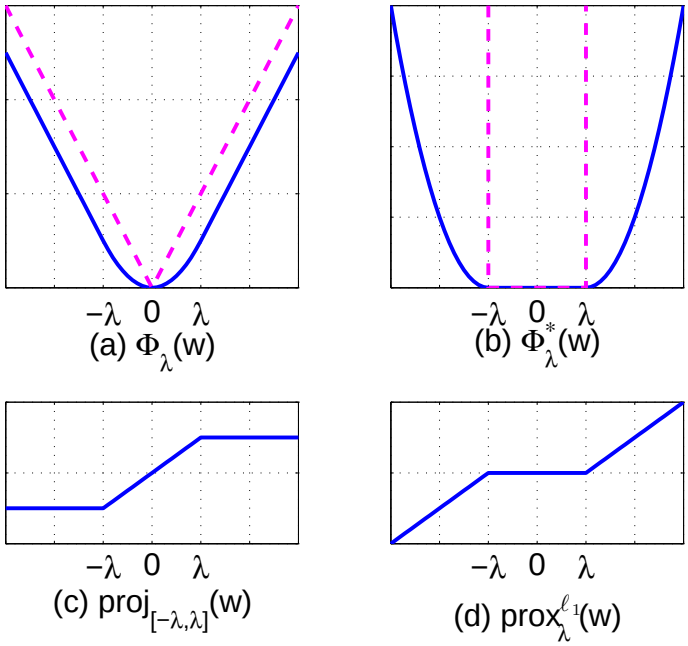



Figure 9: (a) The $\ell_1$-regularizer (dashed) and its envelope function $\Phi_\lambda$ (solid). (b) The indicator function $\delta^\infty_\lambda$ (dashed) and its envelope function $\Phi^*_\lambda$ (solid). (c) The derivative of $\Phi_\lambda$, which is the projection onto the interval $[-\lambda,\lambda]$; see Equation (8). (d) The derivative of $\Phi^*_\lambda$, which is called the soft-threshold function (9). Note that the $\ell_1$-regularizer and the indicator function $\delta^\infty_\lambda$ are conjugate to each other.

which implies the uniqueness of the subgradient of $F(\mathbf{z})$ for all $\mathbf{z}$.

Inequality (65) follows easily from the definition of the envelope function $F$ and Lemma 8 as follows:

$$\begin{aligned} F(\mathbf{z}') - F(\mathbf{z}) &= f(\mathbf{x}') + \frac{1}{2}\|\mathbf{y}'\|^2 - f(\mathbf{x}) - \frac{1}{2}\|\mathbf{y}\|^2 \\ &= \Big(f(\mathbf{x}') - f(\mathbf{x})\Big) + \Big(\frac{1}{2}\|\mathbf{y}'\|^2 - \frac{1}{2}\|\mathbf{y}\|^2\Big) \\ &\geq (\mathbf{x}' - \mathbf{x})^\top \mathbf{y} + (\mathbf{y}' - \mathbf{y})^\top \mathbf{y} \\ &= (\mathbf{z}' - \mathbf{z})^\top \mathbf{y}, \end{aligned}$$

where $\mathbf{x} = \mathrm{prox}_f(\mathbf{z})$, $\mathbf{y} = \mathrm{prox}_{f^*}(\mathbf{z})$, and $\mathbf{x}'$ and $\mathbf{y}'$ are similarly defined. We used the convexity of $f$ with $\mathbf{y} \in \partial f(\mathbf{x})$ and the convexity of $\|\cdot\|^2/2$ in the third line.

Second, we obtain inequality (66) by upper-bounding $F(\mathbf{z}')$ as follows:

$$\begin{aligned}
F(\mathbf{z}') &= \min_{\xi\in\mathbb{R}^n}\Big(f(\xi)+\frac{1}{2}\|\xi-\mathbf{z}'\|^2\Big)\\
&\le f(\mathbf{x})+\frac{1}{2}\|\mathbf{x}-\mathbf{z}'\|^2\\
&= F(\mathbf{z})-\frac{1}{2}\|\mathbf{x}-\mathbf{z}\|^2+\frac{1}{2}\|\mathbf{x}-\mathbf{z}'\|^2\\
&= F(\mathbf{z})-\frac{1}{2}\|\mathbf{x}-\mathbf{z}\|^2+\frac{1}{2}\|\mathbf{x}-\mathbf{z}+\mathbf{z}-\mathbf{z}'\|^2\\
&= F(\mathbf{z})+(\mathbf{x}-\mathbf{z})^\top(\mathbf{z}-\mathbf{z}')+\frac{1}{2}\|\mathbf{z}'-\mathbf{z}\|^2\\
&= F(\mathbf{z})+\mathbf{y}^\top(\mathbf{z}'-\mathbf{z})+\frac{1}{2}\|\mathbf{z}'-\mathbf{z}\|^2.
\end{aligned}$$

■

The envelope functions of two convex functions $\phi_\lambda(w)=\lambda|w|$ and $\phi^*_\lambda(w)=\delta^\infty_\lambda(w)$, and their derivatives (the projection (8) and the soft-threshold function (9), respectively) are schematically shown in Figure 9.

## Appendix B. Derivation of Equation (20)

$$\begin{aligned}
\text{Equation (18)} &= \max_{\alpha\in\mathbb{R}^m}\Bigg\{-f^*_\ell(-\alpha)+\min_{\mathbf{w}\in\mathbb{R}^n}\Big(\phi_\lambda(\mathbf{w})+\frac{1}{2\eta_t}\|\mathbf{w}-\mathbf{w}^t-\eta_t\mathbf{A}^\top\alpha\|^2\Big)\\
&\qquad\qquad -\frac{1}{2\eta_t}\|\mathbf{w}^t+\eta_t\mathbf{A}^\top\alpha\|^2\Bigg\}+\frac{\|\mathbf{w}^t\|^2}{2\eta_t}\\
&= \max_{\alpha\in\mathbb{R}^m}\Big(-f^*_\ell(-\alpha)+\frac{1}{\eta_t}\Phi_{\lambda\eta_t}(\mathbf{w}^t+\eta_t\mathbf{A}^\top\alpha)-\frac{1}{2\eta_t}\|\mathbf{w}^t+\eta_t\mathbf{A}^\top\alpha\|^2\Big)+\frac{\|\mathbf{w}^t\|^2}{2\eta_t}\\
&= \max_{\alpha\in\mathbb{R}^m}\Big(-f^*_\ell(-\alpha)-\frac{1}{\eta_t}\Phi^*_{\lambda\eta_t}(\mathbf{w}^t+\eta_t\mathbf{A}^\top\alpha)\Big)+\frac{\|\mathbf{w}^t\|^2}{2\eta_t},
\end{aligned}$$

where we used the definition of the Moreau envelope in the second line and Lemma 9 in the third line. Finally omitting the constant term $\|\mathbf{w}^t\|^2/(2\eta_t)$ in the last line and reversing the sign we obtain Equation (20).

## Appendix C. Proofs

In this section, we present the proofs of Theorem 3, Lemma 4, Theorem 6, and Theorem 7.

### C.1 Proof of Theorem 3

**Proof** The first step of the proof is to generalize Lemma 1 in two ways: first we allow a point $\mathbf{w}^*$ in the set of minimizers $W^*$ to be chosen for each time step, and second, we introduce a parameter $\mu$ to

tighten the bound. Let $\bar{\mathbf{w}}^t$ be the closest point from $\mathbf{w}^t$ in $W^*$, namely $\bar{\mathbf{w}}^t := \mathrm{argmin}_{\mathbf{w}^* \in W^*} \|\mathbf{w}^t - \mathbf{w}^*\|$. From the proof of Lemma 1, we have

$$\begin{aligned}
\eta_t(f(\bar{\mathbf{w}}^{t+1}) - f(\mathbf{w}^{t+1})) &\geq \left\langle \bar{\mathbf{w}}^{t+1} - \mathbf{w}^{t+1}, \mathbf{w}^t - \mathbf{w}^{t+1} \right\rangle \\
&= \left\langle \bar{\mathbf{w}}^{t+1} - \mathbf{w}^{t+1}, \mathbf{w}^t - \bar{\mathbf{w}}^t + \bar{\mathbf{w}}^t - \bar{\mathbf{w}}^{t+1} + \bar{\mathbf{w}}^{t+1} - \mathbf{w}^{t+1} \right\rangle \\
&= \|\bar{\mathbf{w}}^{t+1} - \mathbf{w}^{t+1}\|^2 + \left\langle \bar{\mathbf{w}}^{t+1} - \mathbf{w}^{t+1}, \mathbf{w}^t - \bar{\mathbf{w}}^t \right\rangle + \left\langle \bar{\mathbf{w}}^{t+1} - \mathbf{w}^{t+1}, \bar{\mathbf{w}}^t - \bar{\mathbf{w}}^{t+1} \right\rangle \\
&\geq \|\mathbf{w}^{t+1} - W^*\|^2 - \|\mathbf{w}^{t+1} - W^*\| \|\mathbf{w}^t - W^*\| \\
&\geq \|\mathbf{w}^{t+1} - W^*\|^2 - \left( \frac{\mu}{2} \|\mathbf{w}^{t+1} - W^*\|^2 + \frac{1}{2\mu} \|\mathbf{w}^t - W^*\|^2 \right) \quad (\forall \mu > 0) \\
&= \left(1 - \frac{\mu}{2}\right) \|\mathbf{w}^{t+1} - W^*\|^2 - \frac{1}{2\mu} \|\mathbf{w}^t - W^*\|^2, \qquad (\star)
\end{aligned}$$

where the last inner product $\left\langle \bar{\mathbf{w}}^{t+1} - \mathbf{w}^{t+1}, \bar{\mathbf{w}}^t - \bar{\mathbf{w}}^{t+1} \right\rangle$ in the third line is non-negative because the set of minimizers $W^*$ is a convex set, and $\bar{\mathbf{w}}^{t+1}$ is the projection of $\mathbf{w}^{t+1}$ onto $W^*$; see Bertsekas (1999, Proposition B.11). In addition, the fifth line follows from the inequality of arithmetic and geometric means.

Note that by setting $\mu = 1$ in $(\star)$ and $\bar{\mathbf{w}}^t = \bar{\mathbf{w}}^{t+1}$, we recover Lemma 1. Now using assumption **(A1)**, we obtain the following expression:

$$\left(2\mu - \mu^2\right) \|\mathbf{w}^{t+1} - W^*\|^2 + 2\mu\sigma\eta_t \|\mathbf{w}^{t+1} - W^*\|^\alpha \leq \|\mathbf{w}^t - W^*\|^2.$$

Maximizing the left hand side with respect to $\mu$, we have $\mu = 1 + \sigma\eta_t \|\mathbf{w}^{t+1} - W^*\|^{\alpha-2}$ and accordingly,

$$\left(1 + \sigma\eta_t \|\mathbf{w}^{t+1} - W^*\|^{\alpha-2}\right)^2 \|\mathbf{w}^{t+1} - W^*\|^2 \leq \|\mathbf{w}^t - W^*\|^2.$$

Taking the square-root of both sides we obtain

$$\|\mathbf{w}^{t+1} - W^*\| + \sigma\eta_t \|\mathbf{w}^{t+1} - W^*\|^{\alpha-1} \leq \|\mathbf{w}^t - W^*\|. \tag{67}$$

The last part of the theorem is obtained by lower-bounding the left-hand side of the above inequality using Young's inequality as follows:

$$\begin{aligned}
&\|\mathbf{w}^{t+1} - W^*\| + \sigma\eta_t \|\mathbf{w}^{t+1} - W^*\|^{\alpha-1} \\
&= (1 + \sigma\eta_t) \left( \frac{1}{1+\sigma\eta_t} \|\mathbf{w}^{t+1} - W^*\| + \frac{\sigma\eta_t}{1+\sigma\eta_t} \|\mathbf{w}^{t+1} - W^*\|^{\alpha-1} \right) \\
&\geq (1 + \sigma\eta_t) \|\mathbf{w}^{t+1} - W^*\|^{\frac{1}{1+\sigma\eta_t}} \cdot \|\mathbf{w}^{t+1} - W^*\|^{\frac{(\alpha-1)\sigma\eta_t}{1+\sigma\eta_t}} \\
&= (1 + \sigma\eta_t) \|\mathbf{w}^{t+1} - W^*\|^{\frac{1+(\alpha-1)\sigma\eta_t}{1+\sigma\eta_t}}.
\end{aligned}$$

Substituting this relation back into inequality (67) completes the proof of the theorem. ■

### C.2 Proof of Lemma 4

**Proof** First let us define $\delta^t \in \mathbb{R}^m$ as the gradient of the AL function (22) at the approximate minimizer $\alpha^t$ follows:

$$\delta^t := \nabla\varphi_t(\alpha^t) = -\nabla f_\ell^*(-\alpha^t) + \mathbf{A}\mathbf{w}^{t+1},$$

where $\mathbf{w}^{t+1} := \mathrm{prox}_{\phi_{\lambda\eta_t}}(\mathbf{w}^t + \eta_t\mathbf{A}^\top\alpha^t)$. Note that $\|\delta^t\| \le \sqrt{\frac{\gamma}{\eta_t}}\|\mathbf{w}^{t+1} - \mathbf{w}^t\|$ from Assumption **(A4)**. Using Corollary 23.5.1 from Rockafellar (1970), we have

$$\nabla f_\ell(\mathbf{A}\mathbf{w}^{t+1} - \delta^t) = \nabla f_\ell\left(\nabla f_\ell^*(-\alpha^t)\right) = -\alpha^t, \tag{68}$$

which implies that if $\delta^t$ is small, $-\alpha^t$ is approximately the gradient of the loss term at $\mathbf{w}^{t+1}$.

Moreover, let $\mathbf{q}^t = \mathbf{w}^t + \eta_t\mathbf{A}^\top\alpha^t$. Since $\mathbf{w}^{t+1} = \mathrm{prox}_{\phi_{\lambda\eta_t}}(\mathbf{q}^t)$ (Assumption **A3**), we have

$$\partial\phi_{\lambda\eta_t}(\mathbf{w}^{t+1}) + (\mathbf{w}^{t+1} - \mathbf{q}^t) \ni 0,$$

which implies

$$(\mathbf{q}^t - \mathbf{w}^{t+1})/\eta_t \in \partial\phi_\lambda(\mathbf{w}^{t+1}), \tag{69}$$

because $\phi_{\lambda\eta_t} = \eta_t\phi_\lambda$.

Now we are ready to derive an analogue of inequality (39) in the proof of Lemma 1. Let $\mathbf{w} \in \mathbb{R}^n$ be an arbitrary vector. We can decompose the residual value in the left hand side of inequality (39) as follows:

$$\begin{aligned}\eta_t(f(\mathbf{w}) - f(\mathbf{w}^{t+1})) = \eta_t(&\underbrace{f_\ell(\mathbf{A}\mathbf{w}) - f_\ell(\mathbf{A}\mathbf{w}^{t+1} - \delta^t)}_{\text{(A)}}) \\ &+ \eta_t(\underbrace{f_\ell(\mathbf{A}\mathbf{w}^{t+1} - \delta^t) - f_\ell(\mathbf{A}\mathbf{w}^{t+1})}_{\text{(B)}}) \\ &+ \eta_t(\underbrace{\phi_\lambda(\mathbf{w}) - \phi_\lambda(\mathbf{w}^{t+1})}_{\text{(C)}}).\end{aligned}$$

The above terms (A), (B), and (C) can be separately bounded using the convexity of $f_\ell$ and $\phi_\lambda$ as follows:

$$\text{(A)}: \qquad f_\ell(\mathbf{A}\mathbf{w}) - f_\ell(\mathbf{A}\mathbf{w}^{t+1} - \delta^t) \ge \left\langle \mathbf{A}(\mathbf{w} - \mathbf{w}^{t+1}) + \delta^t, -\alpha^t \right\rangle,$$

$$\text{(B)}: \qquad f_\ell(\mathbf{A}\mathbf{w}^{t+1} - \delta^t) - f_\ell(\mathbf{A}\mathbf{w}^{t+1}) \ge -\left\langle \delta^t, -\alpha^t \right\rangle - \frac{1}{2\gamma}\|\delta^t\|^2,$$

$$\text{(C)}: \qquad \phi_\lambda(\mathbf{w}) - \phi_\lambda(\mathbf{w}^{t+1}) \ge \left\langle \mathbf{w} - \mathbf{w}^{t+1}, \frac{\mathbf{w}^t + \eta_t\mathbf{A}^\top\alpha^t - \mathbf{w}^{t+1}}{\eta_t} \right\rangle,$$

where (A) is due to Equation (68), (B) is due to assumption **(A2)** and Hiriart-Urruty and Lemaréchal (1993, Theorem X.4.2.2), and (C) is due to Equation (69). Combining (A), (B), and (C), we have the following expression:

$$\eta_t(f(\mathbf{w}) - f(\mathbf{w}^{t+1})) \ge \left\langle \mathbf{w}^t - \mathbf{w}^{t+1}, \mathbf{w} - \mathbf{w}^{t+1} \right\rangle - \frac{\eta_t}{2\gamma}\|\delta^t\|^2.$$

Note that the above inequality reduces to Equation (39) if $\|\delta_t\| = 0$ (exact minimization). Using assumption **(A4)**, we obtain

$$\begin{aligned}\eta_t(f(\mathbf{w}) - f(\mathbf{w}^{t+1})) &\geq \left\langle \mathbf{w}^t - \mathbf{w} + \mathbf{w} - \mathbf{w}^{t+1}, \mathbf{w} - \mathbf{w}^{t+1} \right\rangle - \frac{1}{2}\|\mathbf{w}^t - \mathbf{w}^{t+1}\|^2 \\ &= \frac{1}{2}\|\mathbf{w} - \mathbf{w}^{t+1}\|^2 - \frac{1}{2}\|\mathbf{w} - \mathbf{w}^t\|^2,\end{aligned}$$

which completes the proof. ■

### C.3 Proof of Theorem 6

**Proof** Let $\delta = (1-\varepsilon)/(\sigma\eta_t)$. Note that $\delta \leq 3/4 < 1$ from the assumption. Following the proof of Lemma 4 with $\mathbf{w} = \bar{\mathbf{w}}^t$, we have

$$\begin{aligned}&\eta_t(f(\bar{\mathbf{w}}^{t+1}) - f(\mathbf{w}^{t+1})) \\ &= \eta_t(f(\bar{\mathbf{w}}^t) - f(\mathbf{w}^{t+1})) \\ &\geq \left\langle \mathbf{w}^t - \mathbf{w}^{t+1}, \bar{\mathbf{w}}^t - \mathbf{w}^{t+1} \right\rangle - \frac{\delta}{2}\|\mathbf{w}^{t+1} - \mathbf{w}^t\|^2 \\ &= \left\langle \mathbf{w}^t - \bar{\mathbf{w}}^t + \bar{\mathbf{w}}^t - \mathbf{w}^{t+1}, \bar{\mathbf{w}}^t - \mathbf{w}^{t+1} \right\rangle - \frac{1}{2}\|\mathbf{w}^{t+1} - \mathbf{w}^t\|^2 + \frac{1-\delta}{2}\|\mathbf{w}^{t+1} - \mathbf{w}^t\|^2 \\ &= \frac{1}{2}\|\bar{\mathbf{w}}^t - \mathbf{w}^{t+1}\|^2 - \frac{1}{2}\|\bar{\mathbf{w}}^t - \mathbf{w}^t\|^2 + \frac{1-\delta}{2}\|\mathbf{w}^{t+1} - \mathbf{w}^t\|^2 \\ &\geq \frac{1}{2}\|\mathbf{w}^{t+1} - W^*\|^2 - \frac{1}{2}\|\mathbf{w}^t - W^*\|^2 \\ &\qquad + \frac{1-\delta}{2}\left(\|\mathbf{w}^{t+1} - W^*\|^2 + \|\mathbf{w}^t - W^*\|^2 + 2\left\langle \mathbf{w}^{t+1} - \bar{\mathbf{w}}^{t+1}, \bar{\mathbf{w}}^t - \mathbf{w}^t \right\rangle\right) \\ &\geq -(1-\delta)\|\mathbf{w}^{t+1} - W^*\|\|\mathbf{w}^t - W^*\| + \left(1 - \frac{\delta}{2}\right)\|\mathbf{w}^{t+1} - W^*\|^2 - \frac{\delta}{2}\|\mathbf{w}^t - W^*\|^2 \\ &\geq -(1-\delta)\left(\frac{1}{2\mu}\|\mathbf{w}^t - W^*\|^2 + \frac{\mu}{2}\|\mathbf{w}^{t+1} - W^*\|^2\right) \\ &\qquad + \left(1 - \frac{\delta}{2}\right)\|\mathbf{w}^{t+1} - W^*\|^2 - \frac{\delta}{2}\|\mathbf{w}^t - W^*\|^2 \quad (\forall \mu > 0),\end{aligned}$$

where we used $\|\bar{\mathbf{w}}^{t+1} - \bar{\mathbf{w}}^t\|^2 \geq 0$, $\left\langle \mathbf{w}^{t+1} - \bar{\mathbf{w}}^{t+1}, \bar{\mathbf{w}}^{t+1} - \bar{\mathbf{w}}^t \right\rangle \geq 0$ and $\left\langle \bar{\mathbf{w}}^{t+1} - \bar{\mathbf{w}}^t, \bar{\mathbf{w}}^t - \mathbf{w}^t \right\rangle \geq 0$ in the sixth line; the seventh line follows from Cauchy-Schwartz inequality; the eighth line follows from the inequality of arithmetic and geometric means.

Applying assumption **(A1)** with $\alpha = 2$ to the above expression, we have

$$\frac{1-\delta}{2\mu}\|\mathbf{w}^t - W^*\|^2 \geq -\frac{1-\delta}{2}\mu\|\mathbf{w}^{t+1} - W^*\|^2 + \left(\left(1 - \tfrac{\delta}{2}\right) + \sigma\eta_t\right)\|\mathbf{w}^{t+1} - W^*\|^2 - \frac{\delta}{2}\|\mathbf{w}^t - W^*\|^2.$$

Multiplying both sides with $\mu/\|\mathbf{w}^{t+1} - W^*\|^2$, we have

$$\frac{1-\delta}{2}\frac{\|\mathbf{w}^t - W^*\|^2}{\|\mathbf{w}^{t+1} - W^*\|^2} \geq -\frac{1-\delta}{2}\mu^2 + \left\{\left(1 - \frac{\delta}{2}\right) + \sigma\eta_t - \frac{\delta}{2}\frac{\|\mathbf{w}^t - W^*\|^2}{\|\mathbf{w}^{t+1} - W^*\|^2}\right\}\mu. \tag{70}$$

Now we have to consider two cases depending on the sign inside the curly brackets. If the sign is negative or zero, we have

$$1-\frac{\delta}{2}+\sigma\eta_t-\frac{\delta}{2}\frac{\|\mathbf{w}^t-W^*\|^2}{\|\mathbf{w}^{t+1}-W^*\|^2}\leq 0,$$

which implies

$$\|\mathbf{w}^{t+1}-W^*\|^2\leq\frac{\delta}{2-\delta+2\sigma\eta_t}\|\mathbf{w}^t-W^*\|^2. \tag{71}$$

Since $\delta\leq 3/4$, the factor in front of $\|\mathbf{w}^t-W^*\|^2$ in the right-hand side is clearly smaller than one. We further show that this factor is smaller than $1/(1+\varepsilon\sigma\eta_t)^2$. First we upper bound $\delta$ by $1/(1+\varepsilon\sigma\eta_t)$ as follows:

$$\delta=\frac{1-\varepsilon}{\sigma\eta_t}=\frac{(1-\varepsilon)(\frac{1}{\sigma\eta_t}+\varepsilon)}{1+\varepsilon\sigma\eta_t}\leq\frac{(1-\varepsilon)\varepsilon+3/4}{1+\varepsilon\sigma\eta_t}\leq\frac{1}{1+\varepsilon\sigma\eta_t}.$$

Plugging the above upper bound into inequality (71), we have

$$\begin{aligned}\|\mathbf{w}^{t+1}-W^*\|^2&\leq\frac{\delta}{1+2\sigma\eta_t}\|\mathbf{w}^t-W^*\|^2\\&\leq\frac{1}{(1+\varepsilon\sigma\eta_t)(1+2\sigma\eta_t)}\|\mathbf{w}^t-W^*\|^2\leq\frac{1}{(1+\varepsilon\sigma\eta_t)^2}\|\mathbf{w}^t-W^*\|^2,\end{aligned}$$

which completes the proof for the first case.

If on the other hand, the term inside the curly brackets is positive in Equation (70), maximizing the right-hand side of Equation (70) with respect to $\mu$ gives the following expression:

$$(1-\delta)r_t\geq 1-\frac{\delta}{2}+\sigma\eta_t-\frac{\delta}{2}r_t^2,$$

where we defined $r_t:=\|\mathbf{w}^t-W^*\|/\|\mathbf{w}^{t+1}-W^*\|$. Because $r_t>0$, the above inequality translates into

$$\begin{aligned}r_t&\geq\frac{\sqrt{1+2\sigma\eta_t\delta}-1+\delta}{\delta}\\&\geq\frac{1+\sigma\eta_t\delta-\sigma^2\eta_t^2\delta^2-1+\delta}{\delta}\\&\geq 1+\sigma\eta_t(1-\sigma\eta_t\delta)\\&=1+\varepsilon\sigma\eta_t.\end{aligned}$$

The second line is true because for $x\geq 0$, $\sqrt{1+x}\geq 1+x/2-x^2/4$; the last line follows from the definition $\delta=(1-\varepsilon)/(\sigma\eta_t)$. ■

### C.4 Proof of Theorem 7

**Proof** Since the minimizer is unique, we denote the minimizer by $\mathbf{w}^*$ and show that the following is true:

$$f(\mathbf{w}) - f(\mathbf{w}^*) \geq \sigma \|\mathbf{w} - \mathbf{w}^*\|^2 \qquad (\forall \mathbf{w} : \|\mathbf{w} - \mathbf{w}^*\| \leq c\tau L).$$

Using Theorem X.4.2.2 in Hiriart-Urruty and Lemaréchal (1993), for $\|\beta\| \leq \tau$, we have

$$\begin{aligned} f^*(\beta) &\leq f^*(0) + \beta^\top \nabla f^*(0) + \frac{L}{2}\|\beta\|^2 \\ &= -f(\mathbf{w}^*) + \beta^\top \mathbf{w}^* + \frac{L}{2}\|\beta\|^2, \end{aligned} \tag{72}$$

where $\mathbf{w}^* := \operatorname{argmin}_{\mathbf{w}\in\mathbb{R}^n} f(\mathbf{w}) = \nabla f^*(0)$ and $f^*(0) = -f(\mathbf{w}^*)$.

On the other hand, we have

$$\begin{aligned} f(\mathbf{w}) &= \sup_{\beta\in\mathbb{R}^n} \left(\beta^\top \mathbf{w} - f^*(\beta)\right) \\ &\geq \sup_{\|\beta\|\leq\tau} \left(\beta^\top \mathbf{w} - f^*(\beta)\right) \\ &\geq \sup_{\|\beta\|\leq\tau} \left(\beta^\top (\mathbf{w} - \mathbf{w}^*) - \frac{L}{2}\|\beta\|^2\right) + f(\mathbf{w}^*) \\ &\begin{cases} = f(\mathbf{w}^*) + \frac{1}{2L}\|\mathbf{w} - \mathbf{w}^*\|^2 & (\text{if } c \leq 1), \\ \geq f(\mathbf{w}^*) + \frac{2c-1}{2c^2 L}\|\mathbf{w} - \mathbf{w}^*\|^2 & (\text{otherwise}), \end{cases} \end{aligned}$$

where we used inequality (72) in the third line; the last line follows because if $c \leq 1$, the maximum is attained at $\beta = (\mathbf{w} - \mathbf{w}^*)/L$, and otherwise we can lower bound the value at the maximum by the value at $\beta = (\mathbf{w} - \mathbf{w}^*)/(cL)$. Combining the above two cases, we have Theorem 7. ■

## References

G. Andrew and J. Gao. Scalable training of L1-regularized log-linear models. In *Proc. of the 24th international conference on Machine learning*, pages 33–40, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-793-3. doi: http://doi.acm.org/10.1145/1273496.1273501.

A. Argyriou, T. Evgeniou, and M. Pontil. Multi-task feature learning. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in NIPS 19*, pages 41–48. MIT Press, Cambridge, MA, 2007.

A. Argyriou, C. A. Micchelli, M. Pontil, and Y. Ying. A spectral regularization framework for multi-task structure learning. In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 25–32. MIT Press, Cambridge, MA, 2008.

S. E Baranzini, P. Mousavi, J. Rio, S. J. Caillier, A. Stillman, P. Villoslada, M. M. Wyatt, M. Comabella, L. D. Greller, R. Somogyi, X. Montalban, and J. R. Oksenberg. Transcription-based prediction of response to ifnβ using supervised computational methods. *PLoS Biol.*, 3(1):e2, 2004.

A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sciences*, 2(1):183–202, 2009.

D. P. Bertsekas. *Constrained Optimization and Lagrange Multiplier Methods*. Academic Press, 1982.

D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1999. 2nd edition.

P.J. Bickel, Y. Ritov, and A.B. Tsybakov. Simultaneous analysis of lasso and dantzig selector. *The Annals of Statistics*, 37(4):1705–1732, 2009. ISSN 0090-5364.

J. M. Bioucas-Dias. Bayesian wavelet-based image deconvolution: A GEM algorithm exploiting a class of heavy-tailed priors. *IEEE Trans. Image Process.*, 15:937–951, 2006.

J. M. Bioucas-Dias and M. A. T. Figueiredo. A new twist: two-step iterative shrinkage/thresholding algorithms for image restoration. *IEEE Trans. Image Process.*, 16(12):2992–3004, 2007.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers, 2011. Unfinished working draft.

R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 16(5):1190–1208, 1995. ISSN 1064-8275.

J.-F. Cai, E. J. Candes, and Z. Shen. A singular value thresholding algorithm for matrix completion. Technical report, arXiv:0810.3286, 2008. URL `http://arxiv.org/abs/0810.3286`.

P. L. Combettes and V. R. Wajs. Signal recovery by proximal forward-backward splitting. *Multiscale Modeling and Simulation*, 4(4):1168–1200, 2005.

I. Daubechies, M. Defrise, and C. De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Commun. Pur. Appl. Math.*, LVII:1413–1457, 2004.

D. L. Donoho. De-noising by soft-thresholding. *IEEE Trans. Inform. Theory*, 41(3):613–627, 1995.

J. Duchi and Y. Singer. Efficient online and batch learning using forward backward splitting. *J. Mach. Learn. Res.*, 10:2899–2934, 2009. ISSN 1532-4435.

B. Efron, T. Hastie, R. Tibshirani, and I. Johnstone. Least angle regression. *Annals of Statistics*, 32 (2):407–499, 2004.

M. Fazel, H. Hindi, and S. P. Boyd. A rank minimization heuristic with application to minimum order system approximation. In *Proc. of the American Control Conference*, 2001.

M. A. T. Figueiredo and R. Nowak. An EM algorithm for wavelet-based image restoration. *IEEE Trans. Image Process.*, 12:906–916, 2003.

M. A. T. Figueiredo, J. M. Bioucas-Dias, and R. D. Nowak. Majorization-minimization algorithm for wavelet-based image restoration. *IEEE Trans. Image Process.*, 16(12), 2007a.

M. A. T. Figueiredo, R. D. Nowak, and S. J. Wright. Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems. *IEEE Journal on selected topics in Signal Processing*, 1(4):586–597, 2007b.

M. Girolami. A variational method for learning sparse and overcomplete representations. *Neural Computation*, 13(11):2517–2532, 2001.

T. Goldstein and S. Osher. The split Bregman method for L1 regularized problems. *SIAM Journal on Imaging Sciences*, 2(2):323–343, 2009. ISSN 1936-4954.

I. F. Gorodnitsky and B. D. Rao. Sparse signal reconstruction from limited data using FOCUSS: A re-weighted minimum norm algorithm. *IEEE Trans. Signal Process.*, 45(3), 1997.

I. Guyon, S. Gunn, M. Nikravesh, and L. Zadeh, editors. *Feature Extraction: Foundations and Applications*. Springer Verlag, 2006.

S. Haufe, R. Tomioka, G. Nolte, K.-R. Müller, and M. Kawanabe. Modeling sparse connectivity between underlying brain sources for EEG/MEG. *IEEE Trans. Biomed. Eng.*, 57(8):1954–1963, 2010. ISSN 0018-9294.

M. R. Hestenes. Multiplier and gradient methods. *J. Optim. Theory Appl.*, 4:303–320, 1969.

J.-B. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms II: Advanced Theory and Bundle Methods*. Springer, 1993.

T. S. Jaakkola. *Variational methods for inference and estimation in graphical models*. PhD thesis, Massachusetts Institute of Technology, 1997.

S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinvesky. An interior-point method for large-scale $\ell_1$-regularized least squares. *IEEE journal of selected topics in signal processing*, 1:606–617, 2007.

K. Koh, S.-J. Kim, and S. Boyd. An interior-point method for large-scale $\ell_1$-regularized logistic regression. *Journal of Machine Learning Research*, 8:1519–1555, 2007.

B. W. Kort and D. P. Bertsekas. Combined primal–dual and penalty methods for convex programming. *SIAM Journal on Control and Optimization*, 14(2):268–294, 1976.

G. Lan. An optimal method for stochastic composite optimization. *Mathematical Programming*, 2010. Under revision.

J. Langford, L. Li, and T. Zhang. Sparse online learning via truncated gradient. *J. Mach. Learn. Res.*, 10:777–801, 2009. ISSN 1532-4435.

Z. Lin, M. Chen, L. Wu, and Y. Ma. The augmented lagrange multiplier method for exact recovery of a corrupted low-rank matrices. *Mathematical Programming*, 2009. submitted.

P. L. Lions and B. Mercier. Splitting algorithms for the sum of two nonlinear operators. *SIAM Journal on Numerical Analysis*, 16(6):964–979, 1979.

N. Meinshausen and P. Bühlmann. High-dimensional graphs and variable selection with the lasso. *The Annals of Statistics*, 34(3):1436–1462, 2006. ISSN 0090-5364.

C. A. Micchelli and M. Pontil. Learning the kernel function via regularization. *Journal of Machine Learning Research*, 6:1099–1125, 2005.

J. J. Moreau. Proximité et dualité dans un espace hilbertien. *Bulletin de la S. M. F.*, 93:273–299, 1965.

Y. Nesterov. Gradient methods for minimizing composite objective function. Technical Report 2007/76, Center for Operations Research and Econometrics (CORE), Catholic University of Louvain, 2007.

J. Nocedal and S. Wright. *Numerical Optimization*. Springer, 1999.

J. Palmer, D. Wipf, K. Kreutz-Delgado, and B. Rao. Variational EM algorithms for non-gaussian latent variable models. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 1059–1066. MIT Press, Cambridge, MA, 2006.

M. J. D. Powell. A method for nonlinear constraints in minimization problems. In R. Fletcher, editor, *Optimization*, pages 283–298. Academic Press, London, New York, 1969.

A. Rakotomamonjy, F. R. Bach, S. Canu, and Y. Grandvalet. SimpleMKL. *JMLR*, 9:2491–2521, 2008.

R. T. Rockafellar. *Convex Analysis*. Princeton University Press, 1970.

R. T. Rockafellar. Monotone operators and the proximal point algorithm. *SIAM Journal on Control and Optimization*, 14:877–898, 1976a.

R. T. Rockafellar. Augmented Lagrangians and applications of the proximal point algorithm in convex programming. *Math. of Oper. Res.*, 1:97–116, 1976b.

S. Shalev-Shwartz and N. Srebro. SVM optimization: inverse dependence on training set size. In *Proc. of the 25th International Conference on Machine Learning*, pages 928–935. ACM, 2008.

S. Shalev-Shwartz, Y. Singer, and N. Srebro. Pegasos: Primal estimated sub-gradient solver for SVM. In *Proc. of the 24th International Conference on Machine Learning*, pages 807–814. ACM, 2007.

N. Srebro, J. D. M. Rennie, and T. S. Jaakkola. Maximum-margin matrix factorization. In *Advances in NIPS 17*, pages 1329–1336. MIT Press, Cambridge, MA, 2005.

R. Tibshirani. Regression shrinkage and selection via the lasso. *J. Roy. Stat. Soc. B*, 58(1):267–288, 1996.

R. Tomioka and K.-R. Müller. A regularized discriminative framework for EEG analysis with application to brain-computer interface. *Neuroimage*, 49(1):415–432, 2010.

R. Tomioka and M. Sugiyama. Dual augmented Lagrangian method for efficient sparse reconstruction. *IEEE Signal Processing Letters*, 16(12):1067–1070, 2009.

R. Tomioka and T. Suzuki. Regularization strategies and empirical Bayesian learning for MKL. Technical report, arXiv:1011.3090v1, 2010.

R. Tomioka, T. Suzuki, M. Sugiyama, and H. Kashima. A fast augmented Lagrangian algorithm for learning low-rank matrices. In *Proc. of the 27th International Conference on Machine Learning*. Omnipress, 2010.

R. Tomioka, K. Hayashi, and H. Kashima. Estimation of low-rank tensors via convex optimization. *SIAM J. Matrix Anal. A.*, 2011a. Submitted.

R. Tomioka, T. Suzuki, and M. Sugiyama. Augmented Lagrangian methods for learning, selecting, and combining features. In Suvrit Sra, Sebastian Nowozin, and Stephen J. Wright, editors, *Optimization for Machine Learning*. MIT Press, 2011b.

D. Wipf and S. Nagarajan. A new view of automatic relevance determination. In *Advances in NIPS 20*, pages 1625–1632. MIT Press, 2008.

S. J. Wright, R. D. Nowak, and M. A. T. Figueiredo. Sparse reconstruction by separable approximation. *IEEE Trans. Signal Process.*, 2009.

J. Yang and Y. Zhang. Alternating direction algorithms for L1-problems in compressive sensing. Technical Report TR09-37, Dept. of Computational & Applied Mathematics, Rice University, 2009.

W. Yin, S. Osher, D. Goldfarb, and J. Darbon. Bregman iterative algorithms for l1-minimization with applications to compressed sensing. *SIAM J. Imaging Sciences*, 1(1):143–168, 2008.

J. Yu, S. V. N. Vishwanathan, S. Günter, and N. N. Schraudolph. A quasi-newton approach to nonsmooth convex optimization problems in machine learning. *The Journal of Machine Learning Research*, 11:1145–1200, 2010.

M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *J. Roy. Stat. Soc. B*, 68(1):49–67, 2006.

X. Zhang, M. Burger, and S. Osher. A unified primal-dual algorithm framework based on bregman iteration. *Journal of Scientific Computing*, 46(1):20–46, 2010.

P. Zhao and B. Yu. On model selection consistency of lasso. *J. Mach. Learn. Res.*, 7:2541–2563, 2006. ISSN 1532-4435.

H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society Series B(Statistical Methodology)*, 67(2):301–320, 2005.