Adaptive ADMM with Spectral Penalty Parameter Selection

Zheng Xu[1], Mário A. T. Figueiredo[2], Tom Goldstein[1]
[1]Department of Computer Science, University of Maryland, College Park, MD
[2]Instituto de Telecomunicações, Instituto Superior Técnico, Universidade de Lisboa, Portugal

Abstract

The *alternating direction method of multipliers* (ADMM) is a versatile tool for solving a wide range of constrained optimization problems. However, its performance is highly sensitive to a penalty parameter, making ADMM often unreliable and hard to automate for a non-expert user. We tackle this weakness of ADMM by proposing a method that adaptively tunes the penalty parameter to achieve fast convergence. The resulting *adaptive ADMM* (AADMM) algorithm, inspired by the successful Barzilai-Borwein spectral method for gradient descent, yields fast convergence and relative insensitivity to the initial stepsize and problem scaling.

1 Introduction

The *alternating direction method of multipliers* (ADMM) is an invaluable element of the modern optimization toolbox. ADMM decomposes complex optimization problems into sequences of simpler subproblems, often solvable in closed form; its simplicity, flexibility, and broad applicability, make ADMM a state-of-the-art solver in machine learning, signal processing, and many other areas [Boyd et al., 2011].

It is well known that the efficiency of ADMM hinges on the careful selection of a *penalty parameter*, which needs to be manually tuned by users for their particular problem instances. In contrast, for gradient descent and proximal-gradient methods, adaptive (*i.e.* automated) stepsize selection rules have been proposed, which essentially dispense with user oversight and dramatically boost performance [Barzilai and Borwein, 1988, Fletcher, 2005, Goldstein et al., 2014b, Wright et al., 2009b, Zhou et al., 2006].

Proceedings of the 20^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2017, Fort Lauderdale, Florida, USA. JMLR: W&CP volume 54.

In this paper, we propose to automate and speed up ADMM by using stepsize selection rules adapted from the gradient descent literature, namely the Barzilai-Borwein "spectral" method for smooth unconstrained problems [Barzilai and Borwein, 1988, Fletcher, 2005]. Since ADMM handles multi-term objectives and linear constraints, it is not immediately obvious how to adopt such rules. The keystone of our approach is to analyze the dual of the ADMM problem, which can be written without constraints. To ensure reliability of the method, we develop a correlation criterion that safeguards it against inaccurate stepsize choices. The resulting *adaptive ADMM* (AADMM) algorithm is fully automated and fairly insensitive to the initial stepsize, as testified for by a comprehensive set of experiments.

2 Background and Related Work

2.1 ADMM

ADMM dates back to the 1970s [Gabay and Mercier, 1976, Glowinski and Marroco, 1975]. Its convergence was shown in the 1990s [Eckstein and Bertsekas, 1992], and convergence rates have been the topic of much recent work, *e.g.*, by Goldstein et al. [2014a], He and Yuan [2015], Nishihara et al. [2015]. In the last decade, ADMM became one of the tools of choice to handle a wide variety of optimization problems in machine learning, signal processing, and many other areas [Boyd et al., 2011].

ADMM tackles problems in the form

$$\begin{aligned} \min_{u\in\mathbb{R}^n, v\in\mathbb{R}^m} \quad & H(u)+G(v), \\ \text{subject to} \quad & Au+Bv=b, \end{aligned} \tag{1}$$

where $H:\mathbb{R}^n\to\bar{\mathbb{R}}$ and $G:\mathbb{R}^m\to\bar{\mathbb{R}}$ are closed, proper, convex functions, $A\in\mathbb{R}^{p\times n}$, $B\in\mathbb{R}^{p\times m}$, and $b\in\mathbb{R}^p$. With $\lambda\in\mathbb{R}^p$ denoting the dual variables (Lagrange multipliers), ADMM has the form

$$u_{k+1} = \arg\min_u H(u) + \frac{\tau_k}{2}\|b - Au - Bv_k + \frac{\lambda_k}{\tau_k}\|_2^2 \tag{2}$$

$$v_{k+1} = \arg\min_v G(v) + \frac{\tau_k}{2}\|b - Au_{k+1} - Bv + \frac{\lambda_k}{\tau_k}\|_2^2 \tag{3}$$

$$\lambda_{k+1} = \lambda_k + \tau_k(b - Au_{k+1} - Bv_{k+1}), \tag{4}$$

where the sequence of *penalties* τ_k is the only free choice, and has a high impact on the algorithm's speed. Our goal is to automate this choice, by adaptively tuning τ_k for optimal performance.

The convergence of the algorithm can be monitored using primal and dual "residuals," both of which approach zero as the iterates become more accurate, and which are defined as

$$\begin{aligned} r_k &= b - Au_k - Bv_k, \\ d_k &= \tau_k A^T B(v_k - v_{k-1}), \end{aligned} \tag{5}$$

respectively [Boyd et al., 2011]. The iteration is generally stopped when

$$\begin{aligned} \|r_k\|_2 &\leq \epsilon^{tol} \max\{\|Au_k\|_2, \|Bv_k\|_2, \|b\|_2\} \\ \|d_k\|_2 &\leq \epsilon^{tol} \|A^T \lambda_k\|_2, \end{aligned} \tag{6}$$

where $\epsilon^{tol} > 0$ is the stopping tolerance.

2.2 Parameter tuning and adaptation

Relatively little work has been done on automating ADMM, *i.e.*, on adaptively choosing τ_k. In the particular case of a strictly convex quadratic objective, criteria for choosing an optimal constant penalty have been recently proposed by Ghadimi et al. [2015], Raghunathan and Di Cairano [2014]. Lin et al. [2011] proposed a non-increasing sequence for the linearization parameter in "linearized" ADMM; however, they do not address the question of how to choose the penalty parameter in ADMM or its variants.

Residual balancing (RB) [He et al., 2000, Boyd et al., 2011] is the only available adaptive method for general form problems (1); it is based on the following observation: increasing τ_k strengthens the penalty term, yielding smaller primal residuals but larger dual ones; conversely, decreasing τ_k leads to larger primal and smaller dual residuals. As both residuals must be small at convergence, it makes sense to "balance" them, *i.e.*, tune τ_k to keep both residuals of similar magnitude. A simple scheme for this goal is

$$\tau_{k+1} = \begin{cases} \eta\tau_k & \text{if } \|r_k\|_2 > \mu\|d_k\|_2 \\ \tau_k/\eta & \text{if } \|d_k\|_2 > \mu\|r_k\|_2 \\ \tau_k & \text{otherwise,} \end{cases} \tag{7}$$

with $\mu > 1$ and $\eta > 1$ [Boyd et al., 2011]. RB has recently been adapted to distributed optimization [Song et al., 2015] and other primal-dual splitting methods [Goldstein et al., 2015]. ADMM with adaptive penalty is not guaranteed to converge, unless τ_k is fixed after a finite number of iterations [He et al., 2000].

Despite some practical success of the RB idea, it suffers from several flaws. The relative size of the residuals depends on the scaling of the problem; e.g., with the change of variable $u \leftarrow 10u$, problem (1) can be re-scaled so that ADMM produces an equivalent sequence of iterates with residuals of very different magnitudes. Consequently, RB criteria are arbitrary in some cases, and their performance varies wildly with different problem scalings (see Section 4.4). Furthermore, the penalty parameter may adapt slowly if the initial value is far from optimal. Finally, without a careful choice of η and μ, the algorithm may fail to converge unless adaptivity is turned off [He et al., 2000].

2.3 Dual interpretation of ADMM

We now explain the close relationship between ADMM and *Douglas-Rachdord splitting* (DRS) [Eckstein and Bertsekas, 1992, Esser, 2009, Goldstein et al., 2014a], which plays a central role in the proposed approach. The starting observation is that the dual of problem (1) has the form

$$\min_{\zeta \in \mathbb{R}^p} \underbrace{H^*(A^T\zeta) - \langle \zeta, b \rangle}_{\hat{H}(\zeta)} + \underbrace{G^*(B^T\zeta)}_{\hat{G}(\zeta)}, \tag{8}$$

where F^* denotes the Fenchel conjugate of F, defined as $F^*(y) = \sup_x \langle x, y\rangle - F(x)$ [Rockafellar, 1970].

The DRS algorithm solves (8) by generating two sequences $(\zeta_k)_{k\in\mathbb{N}}$ and $(\hat{\zeta}_k)_{k\in\mathbb{N}}$ according to

$$0 \quad \in \quad \frac{\hat{\zeta}_{k+1} - \zeta_k}{\tau_k} + \partial\hat{H}(\hat{\zeta}_{k+1}) + \partial\hat{G}(\zeta_k) \tag{9}$$

$$0 \quad \in \quad \frac{\zeta_{k+1} - \zeta_k}{\tau_k} + \partial\hat{H}(\hat{\zeta}_{k+1}) + \partial\hat{G}(\zeta_{k+1}), \tag{10}$$

where we use the standard notation $\partial F(x)$ for the subdifferential of F evaluated at x [Rockafellar, 1970].

Referring back to ADMM in (2)–(4), and defining $\hat{\lambda}_{k+1} = \lambda_k + \tau_k(b - Au_{k+1} - Bv_k)$, the optimality condition for the minimization in (2) is

$$0 \in \partial H(u_{k+1}) - A^T \underbrace{(\lambda_k + \tau_k(b - Au_{k+1} - Bv_k))}_{\hat{\lambda}_{k+1}}$$

which is equivalent to $A^T\hat{\lambda}_{k+1} \in \partial H(u_{k+1})$, thus[1] $u_{k+1} \in \partial H^*(A^T\hat{\lambda}_{k+1})$. A similar argument using the optimality condition for (3) leads to $v_{k+1} \in \partial G^*(B^T\lambda_{k+1})$. Recalling (8), we arrive at

$$Au_{k+1} - b \in \partial\hat{H}(\hat{\lambda}_{k+1}) \text{ and } Bv_{k+1} \in \partial\hat{G}(\lambda_{k+1}). \tag{11}$$

Using these identities, we finally have

$$\begin{aligned} \hat{\lambda}_{k+1} &= \lambda_k + \tau_k(b - Au_{k+1} - Bv_k) \\ &\in \lambda_k - \tau_k\big(\partial\hat{H}(\hat{\lambda}_{k+1}) + \partial\hat{G}(\lambda_k)\big) \qquad (12) \\ \lambda_{k+1} &= \lambda_k + \tau_k(b - Au_{k+1} - Bv_{k+1}) \\ &\in \lambda_k - \tau_k\big(\partial\hat{H}(\hat{\lambda}_{k+1}) + \partial\hat{G}(\lambda_{k+1})\big), \qquad (13) \end{aligned}$$

[1] An important property relating F and F^* is that $y \in \partial H(x)$ if and only if $x \in \partial H^*(y)$ [Rockafellar, 1970].

showing that the sequences $(\lambda_k)_{k\in\mathbb{N}}$ and $(\hat{\lambda}_k)_{k\in\mathbb{N}}$ satisfy the same conditions (9) and (10) as $(\zeta_k)_{k\in\mathbb{N}}$ and $(\hat{\zeta}_k)_{k\in\mathbb{N}}$, thus proving that ADMM for problem (1) is equivalent to DRS for its dual (8).

2.4 Spectral stepsize selection

The classical gradient descent step for unconstrained minimization of a smooth function $F\colon \mathbb{R}^n \to \mathbb{R}$ has the form $x_{k+1} = x_k - \tau_k \nabla F(x_k)$. Spectral gradient methods, pioneered by Barzilai and Borwein (BB) [Barzilai and Borwein, 1988], adaptively choose the stepsize τ_k to achieve fast convergence.

In a nutshell, the standard (there are variants) BB method sets $\tau_k = 1/\alpha_k$, with α_k chosen such that $\alpha_k I$ mimics the Hessian of F over the last step, seeking a quasi-Newton step. A least squares criterion yields

$$\alpha_k = \underset{\alpha\in\mathbb{R}}{\operatorname{argmin}} \|\nabla F(x_k) - \nabla F(x_{k-1}) - \alpha(x_k - x_{k-1})\|_2^2, \quad (14)$$

which is an estimate of the curvature of F across the previous step of the algorithm. BB gradient methods often dramatically outperform those with constant stepsize [Fletcher, 2005, Zhou et al., 2006] and have been generalized to handle non-differentiable problems via proximal gradient methods [Wright et al., 2009b, Goldstein et al., 2014b, Goldstein and Setzer, 2010]. Finally, notice that (14) is equivalent to approximating the gradient $\nabla F(x_k)$ as a linear function of x_k,

$$\nabla F(x_k) \approx \nabla F(x_{k-1}) + \alpha_k(x_k - x_{k-1}) = \alpha_k\, x_k + a_k, \quad (15)$$

where $a_k = \nabla F(x_{k-1}) - \alpha_k\, x_{k-1}$. The observation that a local linear approximation of the gradient has an optimal parameter equal to the inverse of the BB stepsize will play an important role below.

3 Spectral penalty parameters

Inspired by the BB method, we propose a spectral penalty parameter selection method for ADMM. We first derive a spectral stepsize rule for DRS, and then adapt this rule to ADMM. Finally, we discuss safeguarding rules to prevent unexpected behavior when curvature estimates are inaccurate.

3.1 Spectral stepsize for DRS

Consider the dual problem (8). Following the observation in (15) about the BB method, we approximate $\partial\hat{H}$ and $\partial\hat{G}$ at iteration k as linear functions,

$$\partial\hat{H}(\hat{\zeta}) = \alpha_k\,\hat{\zeta} + \Psi_k \quad \text{and} \quad \partial\hat{G}(\zeta) = \beta_k\,\zeta + \Phi_k, \quad (16)$$

where $\alpha_k > 0$, $\beta_k > 0$ are local curvature estimates of dual functions $\hat{H}$ and $\hat{G}$, respectively, and $\Psi_k, \Phi_k \subset \mathbb{R}^p$. Once we obtain these curvature estimates, we will be able to exploit the following proposition.

Proposition 1 (Spectral DRS)**.** *Suppose the DRS steps* (9)–(10) *are applied to problem* (8)*, where (omitting the subscript k from $\alpha_k, \beta_k, \Psi_k, \Phi_k$ to lighten the notation in what follows)*

$$\partial\hat{H}(\hat{\zeta}) = \alpha\,\hat{\zeta} + \Psi \qquad \textit{and} \qquad \partial\hat{G}(\zeta) = \beta\,\zeta + \Phi.$$

Then, the minimal residual of $\hat{H}(\zeta_{k+1}) + \hat{G}(\zeta_{k+1})$ is obtained by setting $\tau_k = 1/\sqrt{\alpha\,\beta}$.

Proof. Inserting (16) into the DRS step (9)–(10) yields

$$0 \in \frac{\hat{\zeta}_{k+1} - \zeta_k}{\tau} + (\alpha\,\hat{\zeta}_{k+1} + \Psi) + (\beta\,\zeta_k + \Phi), \quad (17)$$

$$0 \in \frac{\zeta_{k+1} - \zeta_k}{\tau} + (\alpha\,\hat{\zeta}_{k+1} + \Psi) + (\beta\,\zeta_{k+1} + \Phi). \quad (18)$$

From (17)–(18), we can explicitly get the update for $\hat{\zeta}_{k+1}$ as

$$\hat{\zeta}_{k+1} = \frac{1 - \beta\,\tau}{1 + \alpha\,\tau}\zeta_k - \frac{a\tau + b\tau}{1 + \alpha\,\tau}, \quad (19)$$

where $a \in \Psi$ and $b \in \Phi$, and for ζ_{k+1} as

$$\zeta_{k+1} = \frac{1}{1 + \beta\,\tau}\zeta_k - \frac{\alpha\,\tau}{1 + \beta\,\tau}\hat{\zeta}_{k+1} - \frac{a\,\tau + b\tau}{1 + \beta\,\tau} \quad (20)$$

$$= \frac{(1 + \alpha\,\beta\,\tau^2)\zeta_k - (a + b)\tau}{(1 + \alpha\,\tau)(1 + \beta\,\tau)}, \quad (21)$$

where the second equality results from using the expression for $\hat{\zeta}_{k+1}$ in (19).

The residual r_{DR} at ζ_{k+1} is simply the magnitude of the subgradient (corresponding to elements $a \in \Psi$ and $b \in \Phi$) of the objective that is given by

$$r_{DR} = \|(\alpha + \beta)\zeta_{k+1} + (a + b)\|_2 \quad (22)$$

$$= \frac{1 + \alpha\,\beta\,\tau^2}{(1 + \alpha\,\tau)(1 + \beta\,\tau)}\|(\alpha + \beta)\zeta_k + (a + b)\|_2, \quad (23)$$

where ζ_{k+1} in (23) was substituted with (21). The optimal stepsize τ_k minimizes the residual

$$\tau_k = \arg\min_\tau r_{\mathrm{DR}} = \arg\max_\tau \frac{(1 + \alpha\,\tau)(1 + \beta\,\tau)}{1 + \alpha\,\beta\,\tau^2} \quad (24)$$

$$= \arg\max_\tau \frac{(\alpha + \beta)\tau}{1 + \alpha\beta\tau^2} = 1/\sqrt{\alpha\beta}. \quad (25)$$

Finally (recovering the iteration subscript k), notice that $\tau_k = (\hat{\alpha}_k\,\hat{\beta}_k)^{1/2}$, where $\hat{\alpha}_k = 1/\alpha_k$ and $\hat{\beta}_k = 1/\beta_k$ are the spectral gradient descent stepsizes for $\hat{H}$ and $\hat{G}$, at $\hat{\zeta}_k$ and ζ_k, respectively. □

Proposition 1 shows how to adaptively choose τ_k: begin by obtaining linear estimates of the subgradients of the two terms in the dual objective (8); the geometric mean of these optimal gradient descent stepsizes is then the optimal DRS stepsize, thus also the optimal ADMM penalty parameter, due to the equivalence shown in Subsection 2.3.

3.2 Spectral stepsize estimation

We now address the estimation of $\hat{\alpha}_k = 1/\alpha_k$ and $\hat{\beta}_k = 1/\beta_k$. These curvature parameters are estimated based on the results from iteration k and an older iteration $k_0 < k$. Noting (11), we define

$$\Delta\hat{\lambda}_k := \hat{\lambda}_k - \hat{\lambda}_{k_0}$$
$$\Delta\hat{H}_k := \partial\hat{H}(\hat{\lambda}_k) - \partial\hat{H}(\hat{\lambda}_{k_0}) = A(u_k - u_{k_0}).$$

Assuming, as above, a linear model for $\partial\hat{H}$, we expect $\Delta\hat{H}_k \approx \alpha\,\Delta\hat{\lambda}_k + a$. As is typical in BB-type methods [Barzilai and Borwein, 1988, Zhou et al., 2006], α is estimated via one of the two least squares problems

$$\min_{\alpha} \|\Delta\hat{H}_k - \alpha\Delta\hat{\lambda}_k\|_2^2 \;\text{ or }\; \min_{\alpha} \|\alpha^{-1}\Delta\hat{H}_k - \Delta\hat{\lambda}_k\|_2^2.$$

The closed form solutions for the corresponding spectral stepsizes $\hat{\alpha}_k = 1/\alpha_k$ are, respectively,

$$\hat{\alpha}_k^{\mathrm{SD}} = \frac{\langle\Delta\hat{\lambda}_k, \Delta\hat{\lambda}_k\rangle}{\langle\Delta\hat{H}_k, \Delta\hat{\lambda}_k\rangle} \;\text{ and }\; \hat{\alpha}_k^{\mathrm{MG}} = \frac{\langle\Delta\hat{H}_k, \Delta\hat{\lambda}_k\rangle}{\langle\Delta\hat{H}_k, \Delta\hat{H}_k\rangle}, \quad (26)$$

where, following Zhou et al. [2006], SD stands for *steepest descent* and MG for *minimum gradient*. The Cauchy-Schwarz inequality implies that $\hat{\alpha}_k^{\mathrm{SD}} \geq \hat{\alpha}_k^{\mathrm{MG}}$. Rather than choosing one or the other, we suggest the hybrid stepsize rule proposed by Zhou et al. [2006],

$$\hat{\alpha}_k = \begin{cases} \hat{\alpha}_k^{\mathrm{MG}} & \text{if } \; 2\,\hat{\alpha}_k^{\mathrm{MG}} > \hat{\alpha}_k^{\mathrm{SD}} \\ \hat{\alpha}_k^{\mathrm{SD}} - \hat{\alpha}_k^{\mathrm{MG}}/2 & \text{otherwise.} \end{cases} \quad (27)$$

The spectral stepsize $\hat{\beta}_k = 1/\beta_k$ is similarly set to

$$\hat{\beta}_k = \begin{cases} \hat{\beta}_k^{\mathrm{MG}} & \text{if } \; 2\,\hat{\beta}_k^{\mathrm{MG}} > \hat{\beta}_k^{\mathrm{SD}} \\ \hat{\beta}_k^{\mathrm{SD}} - \hat{\beta}_k^{\mathrm{MG}}/2 & \text{otherwise,} \end{cases} \quad (28)$$

where $\hat{\beta}_k^{\mathrm{SD}} = \langle\Delta\lambda_k, \Delta\lambda_k\rangle/\langle\Delta\hat{G}_k, \Delta\lambda_k\rangle$, $\hat{\beta}_k^{\mathrm{MG}} = \langle\Delta\hat{G}_k, \Delta\lambda_k\rangle/\langle\Delta\hat{G}_k, \Delta\hat{G}_k\rangle$, $\Delta\hat{G}_k = B(v_k - v_{k_0})$, and $\Delta\lambda_k = \lambda_k - \lambda_{k_0}$. It is important to note that $\hat{\alpha}_k$ and $\hat{\beta}_k$ are obtained from the iterates of ADMM, *i.e.*, the user is not required to supply the dual problem.

3.3 Safeguarding

On some iterations, the linear models (for one or both subgradients) underlying the spectral stepsize choice may be very inaccurate. When this occurs, the least squares procedure may produce ineffective stepsizes. The classical BB method for unconstrained problems uses a line search to safeguard against unstable stepsizes resulting from unreliable curvature estimates. In ADMM, however, there is no notion of "stable" stepsize (any constant stepsizes is stable), thus line search methods are not applicable. Rather, we propose to safeguard the method by assessing the quality of the curvature estimates, and only updating the stepsize if the curvature estimates satisfy a reliability criterion.

The linear model (16) assumes the change in dual (sub)gradient is linearly proportional to the change in the dual variables. To test the validity of this assumption, we measure the correlation between these quantities (equivalently, the cosine of their angle):

$$\alpha_k^{\mathrm{cor}} = \frac{\langle\Delta\hat{H}_k, \Delta\hat{\lambda}_k\rangle}{\|\Delta\hat{H}_k\|\,\|\Delta\hat{\lambda}_k\|} \;\text{ and }\; \beta_k^{\mathrm{cor}} = \frac{\langle\Delta\hat{G}_k, \Delta\lambda_k\rangle}{\|\Delta\hat{G}_k\|\,\|\Delta\lambda_k\|}. \quad (29)$$

The spectral stepsizes are updated only if the correlations indicate the estimation is credible enough. The safeguarded spectral adaptive penalty rule is

$$\tau_{k+1} = \begin{cases} \sqrt{\hat{\alpha}_k\hat{\beta}_k} & \text{if } \alpha_k^{\mathrm{cor}} > \epsilon^{\mathrm{cor}} \text{ and } \beta_k^{\mathrm{cor}} > \epsilon^{\mathrm{cor}} \\ \hat{\alpha}_k & \text{if } \alpha_k^{\mathrm{cor}} > \epsilon^{\mathrm{cor}} \text{ and } \beta_k^{\mathrm{cor}} \leq \epsilon^{\mathrm{cor}} \\ \hat{\beta}_k & \text{if } \alpha_k^{\mathrm{cor}} \leq \epsilon^{\mathrm{cor}} \text{ and } \beta_k^{\mathrm{cor}} > \epsilon^{\mathrm{cor}} \\ \tau_k & \text{otherwise,} \end{cases} \quad (30)$$

where ϵ^{cor} is a quality threshold for the curvature estimates, while $\hat{\alpha}_k$ and $\hat{\beta}_k$ are the stepsizes given by (27)–(28). Notice that (30) falls back to constant τ_k when both curvature estimates are deemed inaccurate.

3.4 Adaptive ADMM

Algorithm 1 shows the complete *adaptive ADMM* (AADMM). We suggest only updating the stepsize every T_f iterations. Safeguarding threshold $\epsilon^{\mathrm{cor}} = 0.2$ and $T_f = 2$ generally perform well. The overhead of AADMM over ADMM is modest: only a few inner products plus the storage to keep one previous iterate.

Algorithm 1 Adaptive ADMM (AADMM)

Input: initialize v_0, λ_0, τ_0, $k_0 = 0$,

while not converge by (6) **and** $k <$ maxiter **do**
  $u_{k+1} = \arg\min_u H(u) + \frac{\tau_k}{2}\|b - Au - Bv_k + \frac{\lambda_k}{\tau_k}\|_2^2$
  $v_{k+1} = \arg\min_v G(v) + \frac{\tau_k}{2}\|b - Au_{k+1} - Bv + \frac{\lambda_k}{\tau_k}\|_2^2$
  $\lambda_{k+1} \leftarrow \lambda_k + \tau_k(b - Au_{k+1} - Bv_{k+1})$
  if $\mathrm{mod}(k, T_f) = 1$ **then**
    $\hat{\lambda}_{k+1} = \lambda_k + \tau_k(b - Au_{k+1} - Bv_k)$
    Estimate spectral stepsizes $\hat{\alpha}_k, \hat{\beta}_k$ in (27) (28)
    Estimate correlations $\alpha_k^{\mathrm{cor}}, \beta_k^{\mathrm{cor}}$ in (29)
    Update τ_{k+1} in (30)
    $k_0 \leftarrow k$
  else
    $\tau_{k+1} \leftarrow \tau_k$
  end if
  $k \leftarrow k + 1$
end while

3.5 Convergence

He et al. [2000] proved that convergence is guaranteed for ADMM with adaptive penalty when either of the two following conditions are satisfied:
Condition 1 (Bounded increasing)**.**

$$\sum_{k=1}^{\infty}(\eta_k)^2 < \infty, \ \textit{where } \eta_k = \sqrt{\max\{\frac{\tau_k}{\tau_{k-1}}, 1\} - 1}. \quad (31)$$

Condition 2 (Bounded decreasing)**.**

$$\sum_{k=1}^{\infty}(\theta_k)^2 < \infty, \ \textit{where } \theta_k = \sqrt{\max\{\frac{\tau_{k-1}}{\tau_k}, 1\} - 1}. \quad (32)$$

Condition 1 (Condition 2) suggests that increasing (decreasing) of adaptive penalty is bounded. In practice, these conditions can be satisfied by turning off adaptivity after a finite number of steps, which we have found unnecessary in our experiments with AADMM.

4 Experiments

4.1 Experimental setting

We consider several applications to demonstrate the effectiveness of the proposed AADMM. We focus on statistical problems involving non-differentiable objectives: linear regression with elastic net regularization [Efron et al., 2004, Goldstein et al., 2014a], low rank least squares [Yang and Yuan, 2013, Xu et al., 2015], quadratic programming (QP) [Boyd et al., 2011, Ghadimi et al., 2015, Goldstein et al., 2014a, Raghunathan and Di Cairano, 2014], basis pursuit [Boyd et al., 2011, Goldstein et al., 2014a], consensus ℓ_1-regularized logistic regression [Boyd et al., 2011], and semidefinite programming [Burer and Monteiro, 2003, Wen et al., 2010]. We use both synthetic and benchmark datasets (obtained from the UCI repository and the LIBSVM page) used by Efron et al. [2004], Lee et al. [2006], Liu et al. [2009], Schmidt et al. [2007], Wright et al. [2009b], and Zou and Hastie [2005]. For the small and medium sized datasets, the features are standardized to zero mean and unit variance, whereas for the large and sparse datasets the features are scaled to be in $[-1, 1]$.

For comparison, we implemented *vanilla* ADMM (fixed stepsize), fast ADMM with a restart strategy [Goldstein et al., 2014a], and ADMM with residual balancing [Boyd et al., 2011, He et al., 2000], using (7) with $\mu = 10$ and $\eta = 2$, and adaptivity was turned off after 1000 iterations to guarantee convergence. The proposed AADMM is implemented as shown in Algorithm 1, with fixed parameters $\epsilon^{\text{cor}} = 0.2$ and $T_f = 2$.

We set the stopping tolerance to $\epsilon^{tol} = 10^{-5}, 10^{-3}$, and 0.05 for small, medium, and large scale problems, respectively. The initial penalty $\tau_0 = 0.1$ is used for all problems, except the canonical QP, where τ_0 is set to the value proposed for quadratic problems by Raghunathan and Di Cairano [2014]. For each problem, the same randomly generated initial variables v_0, λ_0 are used for ADMM and all the variants thereof.

4.2 Applications

Elastic net (EN) is a modification of ℓ_1-regularized linear regression (a.k.a. LASSO) that helps preserve groups of highly correlated variables [Zou and Hastie, 2005, Goldstein et al., 2014a] and requires solving

$$\min_x \frac{1}{2}\|Dx - c\|_2^2 + \rho_1\|x\|_1 + \frac{\rho_2}{2}\|x\|_2^2, \quad (33)$$

where, as usual, $\|\cdot\|_1$ and $\|\cdot\|_2$ denote the ℓ_1 and ℓ_2 norms, D is a data matrix, c contains measurements, and x is the vector of regression coefficients. One way to apply ADMM to this problem is to rewrite it as

$$\begin{aligned} &\min_{u,v} \frac{1}{2}\|Du - c\|_2^2 + \rho_1\|v\|_1 + \frac{\rho_2}{2}\|v\|_2^2 \\ &\text{subject to} \quad u - v = 0. \end{aligned} \quad (34)$$

The synthetic dataset introduced by Zou and Hastie [2005] and realistic dataset introduced by Efron et al. [2004], Zou and Hastie [2005] are investigated. Typical parameters $\rho_1 = \rho_2 = 1$ are used in all experiments.

Low rank least squares (LRLS) uses the nuclear matrix norm (sum of singular values) as the convex surrogate of matrix rank,

$$\min_X \frac{1}{2}\|DX - C\|_F^2 + \rho_1\|X\|_* + \frac{\rho_2}{2}\|X\|_F^2, \quad (35)$$

where $\|\cdot\|_*$ denotes the nuclear norm, $\|\cdot\|_F$ is the Frobenius norm, $D \in \mathbb{R}^{n\times m}$ is a data matrix, $C \in \mathbb{R}^{n\times d}$ contains measurements, and $X \in \mathbb{R}^{m\times d}$ is the variable matrix. ADMM can be applied after rewriting (35) as Yang and Yuan [2013], Xu et al. [2015]

$$\begin{aligned} &\min_{U,V} \frac{1}{2}\|DU - C\|_F^2 + \rho_1\|V\|_* + \frac{\rho_2}{2}\|V\|_F^2, \\ &\text{subject to} \quad U - V = 0. \end{aligned} \quad (36)$$

A synthetic problem is constructed using a random data matrix $D \in \mathbb{R}^{1000\times 200}$, a low rank matrix $X \in \mathbb{R}^{200\times 500}$, and $C = DW + \text{Noise}$. We use the binary classification problems introduced by Lee et al. [2006] and Schmidt et al. [2007], where each column of X represents a linear exemplar classifier, trained with a positive sample and all negative samples [Xu et al., 2015]; $\rho_1 = \rho_2 = 1$ is used for all experiments.

Support vector machine (SVM) and QP: the

Table 1: Iterations (and runtime in seconds) for the various algorithms and applications described in the text. Absence of convergence after n iterations is indicated as $n+$. AADMM is the proposed Algorithm 1.

Application	Dataset	#samples × #features[1]	Vanilla ADMM	Fast ADMM	Residual balance	**Adaptive ADMM**
Elastic net regression	Synthetic	50 × 40	2000+ (1.64)	263 (.270)	111 (.129)	**43 (.046)**
	Boston	506 × 13	2000+ (2.19)	208 (.106)	54 (.023)	**17 (.011)**
	Diabetes	768 × 8	594 (.269)	947 (.848)	28 (.020)	**10 (.005)**
	Leukemia	38 × 7129	2000+ (22.9)	2000+ (24.2)	1737 (19.3)	**152 (1.70)**
	Prostate	97 × 8	548 (.293)	139 (.049)	29 (.015)	**16 (.012)**
	Servo	130 × 4	142 (.040)	44 (.017)	27 (.012)	**13 (.007)**
Low rank least squares	Synthetic	1000 × 200	543(31.3)	129(7.30)	75(5.59)	**13(.775)**
	Madelon	2000 × 500	1943(925)	193(89.6)	133(60.9)	**27(12.8)**
	Sonar	208 × 60	1933(9.12)	313(1.51)	102(.466)	**31(.160)**
	Splice	1000 × 60	1704(38.2)	189(4.25)	92(2.04)	**18(.413)**
QP and dual SVM	Synthetic	250 × 500	439 (6.15)	535 (7.8380)	232 (3.27)	**71 (.984)**
	Madelon	2000 × 500	100 (14.0)	57 (8.14)	28 (4.12)	**19 (2.64)**
	Sonar	208 × 60	139 (.227)	43 (.075)	37 (.069)	**28 (.050)**
	Splice	1000 × 60	149 (4.9)	47 (1.44)	39 (1.27)	**20 (.681)**
Basis pursuit	Synthetic	10 × 30	163 (.027)	2000+ (.310)	159 (.031)	**114 (.026)**
	Human1	1024 × 1087	2000+ (2.35)	2000+ (2.41)	839 (.990)	**503 (.626)**
	Human2	1024 × 1087	2000+ (2.26)	2000+ (2.42)	875 (1.03)	**448 (.554)**
	Human3	1024 × 1087	2000+ (2.29)	2000+ (2.44)	713 (.855)	**523 (.641)**
Consensus logistic regression	Synthetic	1000 × 25	301 (3.36)	444 (3.54)	43 (.583)	**22 (.282)**
	Madelon	2000 × 500	2000+ (205)	2000+ (166)	115 (42.1)	**23 (20.8)**
	Sonar	208 × 60	2000+ (33.5)	2000+ (47)	106 (2.82)	**90 (1.64)**
	Splice	1000 × 60	2000+ (29.1)	2000+ (43.7)	86 (1.91)	**22 (.638)**
	News20	19996 × 1355191	69 (5.91e3)	32 (3.45e3)	18 (1.52e3)	**16 (1.2e3)**
	Rcv1	20242 × 47236	38 (177)	23 (122)	**13 (53.0)**	**12 (53.9)**
	Realsim	72309 × 20958	1000+ (2.73e3)	1000+ (1.86e3)	121 (558)	**22 (118)**
Semidefinite programming	hamming-7-5-6	128 × 1792	455(1.78)	2000+(8.60)	1093(4.21)	**284(1.11)**
	hamming-8-3-4	256 × 16128	418(6.38)	2000+(29.1)	1071(16.5)	**118(2.02)**
	hamming-9-5-6	512 × 53760	2000+(187)	2000+(187)	1444(131)	**481(53.1)**
	hamming-9-8	512 × 2304	2000+(162)	2000+(159)	1247(97.2)	**594(52.7)**
	hamming-10-2	1024 × 23040	2000+(936)	2000+(924)	1194(556)	**391(193)**
	hamming-11-2	2048 × 56320	2000+(6.43e3)	2000+(6.30e3)	1203(4.15e3)	**447(1.49e3)**

[1] #constrains × #unknowns for canonical QP; #vertices × #edges for SDP.

dual of the SVM learning problem is a QP

$$\begin{aligned} &\min_z \ \frac{1}{2} z^T Q z - e^T z \\ &\text{subject to} \quad c^T z = 0 \ \text{ and } \ 0 \le z \le C, \end{aligned} \tag{37}$$

where z is the SVM dual variable, Q is the kernel matrix, c is a vector of labels, e is a vector of ones, and $C > 0$ [Chang and Lin, 2011]. We also consider the canonical QP

$$\min_x \ \frac{1}{2} x^T Q x + q^T x \quad \text{subject to} \quad Dx \le c, \tag{38}$$

which can be solved by applying ADMM to

$$\begin{aligned} &\min_{u,v} \ \frac{1}{2} u^T Q u + q^T u + \iota_{\{z:\, z_i \le c\}}(v) \\ &\text{subject to} \quad Du - v = 0; \end{aligned} \tag{39}$$

here, ι_S is the indicator function of set S: $\iota_S(v) = 0$, if $v \in S$, and $\iota_S(v) = \infty$, otherwise.

We study classification problems from Lee et al. [2006] and Schmidt et al. [2007] with $C = 1$, and a random synthetic QP [Goldstein et al., 2014a], where $Q \in \mathbb{R}^{500\times 500}$ with condition number $\simeq 4.5 \times 10^5$.

Basis pursuit (BP) seeks a sparse representation of a vector c by solving the constrained problem

$$\min_x \|x\|_1 \quad \text{subject to} \quad Dx = c, \tag{40}$$

where $D \in \mathbb{R}^{m\times n}, c \in \mathbb{R}^m, m < n$. An extended form with $\hat{D} = [D, I] \in \mathbb{R}^{m\times(n+m)}$ has been used to reconstruct occluded and corrupted faces [Wright et al., 2009a]. To apply ADMM, problem (40) is rewritten as

$$\min_{u,v} \iota_{\{z:\, Dz=c\}}(u) + \|v\|_1 \quad \text{subject to} \quad u - v = 0. \tag{41}$$

We experiment with synthetic random $D \in \mathbb{R}^{10\times 30}$. We also use a data matrix for face reconstruction from the Extended Yale B Face dataset [Wright et al., 2009b], where each frontal face image is scaled to 32×32. For each human subject, an image is selected and corrupted with 5% noisy pixels, and the remaining images from the same subject are used to reconstruct the corrupted image.

Consensus ℓ_1-regularized logistic regression is formulated as a distribute optimization problem with

the form

$$\min_{x_i,z} \sum_{i=1}^{N} \sum_{j=1}^{n_i} \log(1+\exp(-c_j D_j^T x_i)) + \rho\|z\|_1 \quad \text{subject to} \quad x_i - z = 0, i = 1,\dots,N, \tag{42}$$

where $x_i \in \mathbb{R}^m$ represents the local variable on the ith distributed node, z is the global variable, n_i is the number of samples in the ith block, $D_j \in \mathbb{R}^m$ is the jth sample, and $c_j \in \{-1, 1\}$ is the corresponding label. The goal of this example is to test AADMM also in distributed/consensus problems, for which ADMM has become an important tool [Boyd et al., 2011].

A synthetic problem is constructed with Gaussian random data and sparse ground truth solutions. Binary classification problems from Lee et al. [2006], Liu et al. [2009], and Schmidt et al. [2007] are also used to test the effectiveness of the proposed method. We use $\rho = 1$, for small and medium datasets, and $\rho = 5$ for the large datasets to encourage sparsity. We split the data equally into two blocks and use a loop to simulate the distributed computing of consensus subproblems.

Semidefinite programming (SDP) solves the problem

$$\min_X \langle F, X\rangle \quad \text{subject to} \quad X \succeq 0, \quad \mathcal{D}(X) = c, \tag{43}$$

where $X \succeq 0$ means that X is positive semidefinite, X, F, $D_i \in \mathbb{R}^{n\times n}$ are symmetric matrices, inner product $\langle X, Y\rangle = \text{trace}(X^T Y)$, and $\mathcal{D}(X) = (\langle D_1, X\rangle, \dots, \langle D_m, X\rangle)^T$. ADMM is applied to the dual form of (43),

$$\min_{y,S} \; -c^T y \quad \text{subject to} \quad \mathcal{D}^*(y) + S = F, \quad S \succeq 0, \tag{44}$$

where $\mathcal{D}^*(y) = \sum_{i=1}^m y_i D_i$, and S is a symmetric positive semidefinite matrix.

As test data, we use 6 graphs from the *Seventh DIMACS Implementation Challenge on Semidefinite and Related Optimization Problems* (following Burer and Monteiro [2003]).

4.3 Convergence results

Table 1 reports the convergence speed of ADMM and its variants for the applications described in Subsection 4.2. Vanilla ADMM with fixed stepsize does poorly in practice: in 13 out of 23 realistic datasets, it fails to converge in the maximum number of iterations. Fast ADMM [Goldstein et al., 2014a] often outperforms vanilla ADMM, but does not compete with the proposed AADMM, which also outperforms residual balancing in all test cases except in the Rcv1 problem for consensus logistic regression.

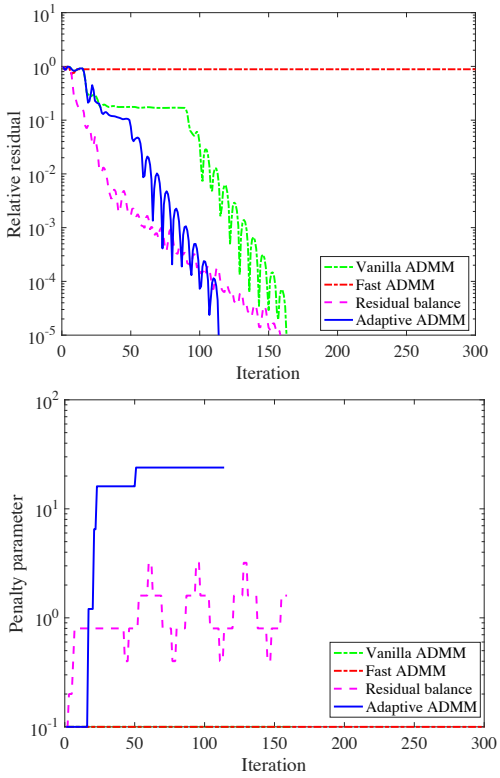

Figure 1: Relative residual (top) and penalty parameter (bottom) for the synthetic basis pursuit (BP) problem.

Fig. 1 presents the relative residual (top) and penalty parameter (bottom) for the synthetic BP problem. The relative residual is defined as

$$\max\left\{\frac{\|r_k\|_2}{\max\{\|Au_k\|_2, \|Bv_k\|_2, \|b\|_2\}}, \frac{\|d_k\|_2}{\|A^T\lambda_k\|_2}\right\},$$

which is based on stopping criterion (6). Fast ADMM often restarts and is slow to converge. The penalty parameter chosen by RB oscillates. AADMM quickly adapts the penalty parameter and converges fastest.

4.4 Sensitivity

We study the sensitivity of the different ADMM variants to problem scaling and initial penalty parameter (τ_0). Scaling sensitivity experiments were done by multiplying the measurement vector c by a scalar s. Fig. 2 presents iteration counts for a wide range of values of initial penalty τ_0 (top) and problems scale s (bottom) for EN regression, canonical QP, and LRLS with synthetic datasets. Fast ADMM and vanilla ADMM use the fixed initial penalty parameter τ_0, and are highly sensitive to this choice, as shown in Fig. 2; in contrast, AADMM is very stable with respect to τ_0 and the scale s.

Finally, Fig. 3 presents iteration counts when applying AADMM with various safeguarding correlation thresholds ϵ^{cor}. When $\epsilon^{\text{cor}} = 0$ the new penalty value is al-

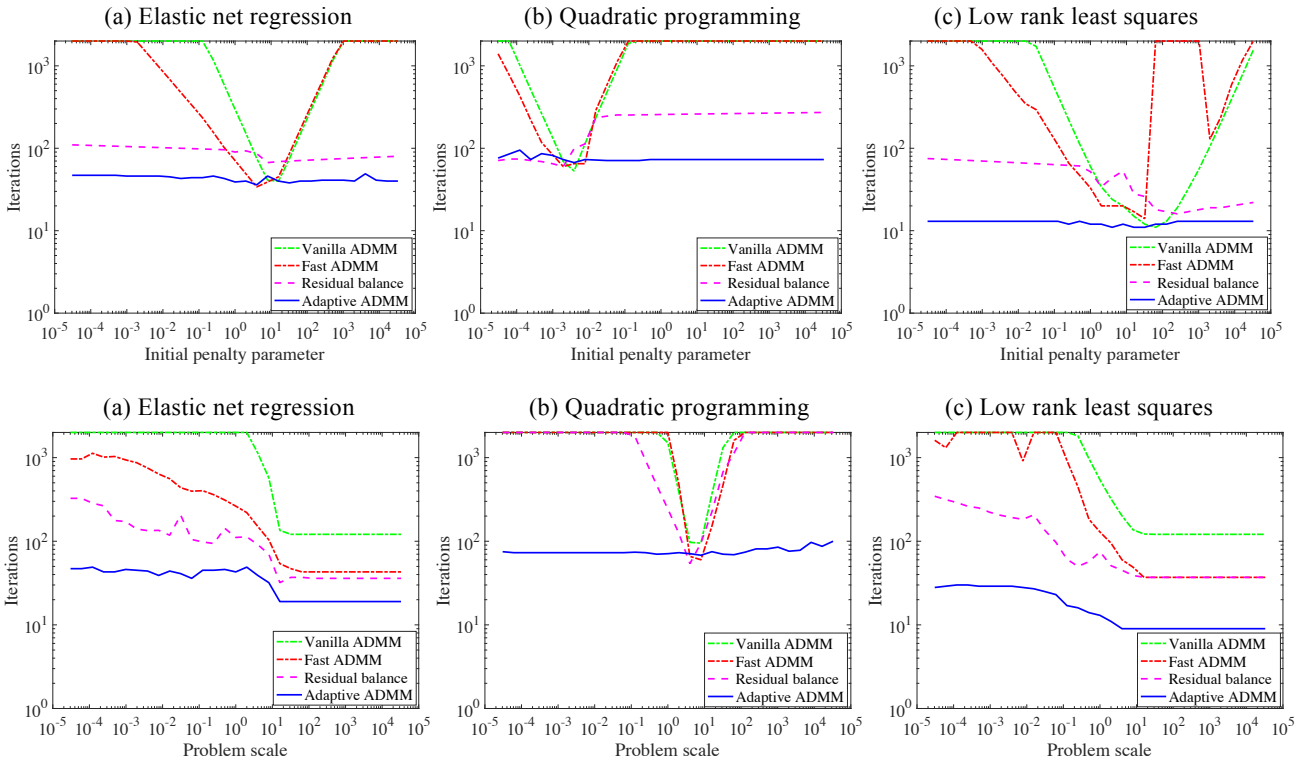

Figure 2: Top row: sensitivity of convergence speed to initial penalty parameter τ_0 for EN, QP, and LRLS. Bottom row: sensitivity to problem scaling s for EN, QP, and LRLS.

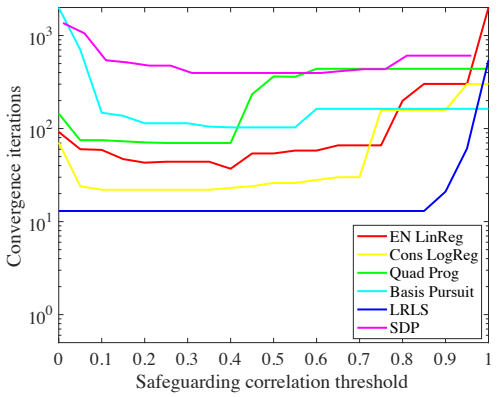

Figure 3: Sensitivity of convergence speed to safeguarding threshold ϵ^{cor} for proposed AADMM. Synthetic problems of various applications are studied. Best viewed in color.

ways accepted, and when $\epsilon^{\text{cor}} = 1$ the penalty parameter is never changed. The proposed AADMM method is insensitive to ϵ^{cor} and performs well for a wide range of $\epsilon^{\text{cor}} \in [0.1, 0.4]$ for various applications.

5 Conclusion

We have proposed *adaptive ADMM* (AADMM), a new variant of the popular ADMM algorithm that tackles one of its fundamental drawbacks: critical dependence on a penalty parameter that needs careful tuning. This drawback has made ADMM difficult to use by non-experts, thus AADMM has the potential to contribute to wider and easier applicability of this highly flexible and efficient optimization tool. Our approach imports and adapts the Barzilai-Borwein "spectral" stepsize method from the smooth optimization literature, tailoring it to the more general class of problems handled by ADMM. The cornerstone of our approach is the fact that ADMM is equivalent to Douglas-Rachford splitting (DRS) applied to the dual problem, for which we develop a spectral stepsize selection rule; this rule is then translated into a criterion to select the penalty parameter of ADMM. A safeguarding function that avoids unreliable stepsize choices finally yields AADMM. Experiments on a comprehensive range of problems and datasets have shown that AADMM outperforms other variants of ADMM and is robust with respect to initial parameter choice and problem scaling.

Acknowledgments

TG and ZX were supported by the US Office of Naval Research (N00014-17-1-2078), and by the US National Science Foundation (CCF-1535902). MF was partially supported by the Fundação para a Ciência e Tecnologia, grant UID/EEA/5008/2013.

References

J. Barzilai and J. Borwein. Two-point step size gradient methods. *IMA J. Num. Analysis*, 8:141–148, 1988.

S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3:1–122, 2011.

S. Burer and R. Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Mathematical Programming*, 95 (2):329–357, 2003.

C.-C. Chang and C.-J. Lin. LIBSVM: a library for support vector machines. *ACM Trans. on Intelligent Systems and Technology*, 2(3):27, 2011.

J. Eckstein and D. Bertsekas. On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical Programming*, 55(1-3):293–318, 1992.

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

E. Esser. Applications of Lagrangian-based alternating direction methods and connections to split Bregman. *CAM report*, 9:31, 2009.

R. Fletcher. On the Barzilai-Borwein method. In *Optimization and control with applications*, pages 235–256. Springer, 2005.

D. Gabay and B. Mercier. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers & Mathematics with Applications*, 2(1):17–40, 1976.

E. Ghadimi, A. Teixeira, I. Shames, and M. Johansson. Optimal parameter selection for the alternating direction method of multipliers: quadratic problems. *IEEE Trans. Autom. Control*, 60:644–658, 2015.

R. Glowinski and A. Marroco. Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problémes de Dirichlet non linéaires. *ESAIM: Modlisation Mathmatique et Analyse Numrique*, 9:41–76, 1975.

T. Goldstein and S. Setzer. High-order methods for basis pursuit. *UCLA CAM Report*, pages 10–41, 2010.

T. Goldstein, B. O'Donoghue, S. Setzer, and R. Baraniuk. Fast alternating direction optimization methods. *SIAM Jour. Imaging Sci.*, 7:1588–1623, 2014a.

T. Goldstein, C. Studer, and R. Baraniuk. A field guide to forward-backward splitting with a FASTA implementation. *arXiv:1411.3406*, 2014b.

T. Goldstein, M. Li, and X. Yuan. Adaptive primal-dual splitting methods for statistical learning and image processing. In *Advances in Neural Information Processing Systems*, pages 2080–2088, 2015.

B. He and X. Yuan. On non-ergodic convergence rate of Douglas-Rachford alternating direction method of multipliers. *Numerische Math.*, 130:567–577, 2015.

B. He, H. Yang, and S. Wang. Alternating direction method with self-adaptive penalty parameters for monotone variational inequalities. *Jour. Optim. Theory and Appl.*, 106(2):337–356, 2000.

S.-I. Lee, H. Lee, P. Abbeel, and A. Ng. Efficient L1 regularized logistic regression. In *AAAI*, volume 21, page 401, 2006.

Z. Lin, R. Liu, and Z. Su. Linearized alternating direction method with adaptive penalty for low-rank representation. In *NIPS*, pages 612–620, 2011.

J. Liu, J. Chen, and J. Ye. Large-scale sparse logistic regression. In *ACM SIGKDD*, pages 547–556, 2009.

R. Nishihara, L. Lessard, B. Recht, A. Packard, and M. Jordan. A general analysis of the convergence of ADMM. In *ICML*, 2015.

A. Raghunathan and S. Di Cairano. Alternating direction method of multipliers for strictly convex quadratic programs: Optimal parameter selection. In *American Control Conf.*, pages 4324–4329, 2014.

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

M. Schmidt, G. Fung, and R. Rosales. Fast optimization methods for l1 regularization: A comparative study and two new approaches. In *ECML*, pages 286–297. Springer, 2007.

C. Song, S. Yoon, and V. Pavlovic. Fast ADMM algorithm for distributed optimization with adaptive penalty. *arXiv:1506.08928*, 2015.

Z. Wen, D. Goldfarb, and W. Yin. Alternating direction augmented lagrangian methods for semidefinite programming. *Mathematical Programming Computation*, 2(3-4):203–230, 2010.

J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 31:210–227, 2009a.

S. Wright, R Nowak, and M. Figueiredo. Sparse reconstruction by separable approximation. *IEEE Trans. Signal Processing*, 57:2479–2493, 2009b.

Z. Xu, X. Li, K. Yang, and T. Goldstein. Exploiting low-rank structure for discriminative subcategorization. In *Proceedings of BMVC, Swansea, UK, September 7-10, 2015*, 2015.

J. Yang and X. Yuan. Linearized augmented lagrangian and alternating direction methods for nuclear norm minimization. *Math. of Computation*, 82: 301–329, 2013.

B. Zhou, L. Gao, and Y.-H. Dai. Gradient methods with adaptive step-sizes. *Computational Optimization and Applications*, 35:69–86, 2006.

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