Adaptive ADMM with Spectral Penalty Parameter Selection

Zheng Xu\textsuperscript{1}, Mário A. T. Figueiredo\textsuperscript{2}, Tom Goldstein\textsuperscript{1}
\textsuperscript{1}Department of Computer Science, University of Maryland, College Park, MD
\textsuperscript{2}Instituto de Telecomunicações, Instituto Superior Técnico, Universidade de Lisboa, Portugal
\textsuperscript{1}\{xuzh,tomg\}@cs.umd.edu, \textsuperscript{2}mario.figueiredo@tecnico.ulisboa.pt

Abstract

The alternating direction method of multipliers (ADMM) is a versatile tool for solving a wide range of constrained optimization problems, with differentiable or non-differentiable objective functions. Unfortunately, its performance is highly sensitive to a penalty parameter, which makes ADMM often unreliable and hard to automate for a non-expert user. We tackle this weakness of ADMM by proposing a method to adaptively tune the penalty parameters 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, made ADMM a state-of-the-art solver in machine learning, signal processing, and many other areas [2].

It is well known that the efficiency of ADMM hinges on the careful selection of a penalty parameter, which is often manually tuned by users for their particular problem instances. 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 [1, 7, 12, 23, 24].

In this paper, we propose to automate and speed up ADMM by using stepsize selection rules imported from the gradient descent literature, namely the Barzilai-Borwein “spectral” method for smooth unconstrained problems [1,7]. Since ADMM handles multi-term objectives and linear constraints, it is not immediately obvious how to adopt such rules. The key to 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) is fully automated and fairly insensitive to the initial stepsize.

2 Background and Related Work

2.1 Alternating Direction Method of Multipliers

ADMM dates back to the 1970s [8,10]. Its convergence was shown in the 1990s [4], and convergence rates have been the topic of much recent work (see [11,14,18] and references therein). 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 (for a comprehensive review, see [2]).

ADMM tackles problems in the form
\[
\min_{u \in \mathbb{R}^n, v \in \mathbb{R}^m} H(u) + G(v), \text{ subject to } Au + Bv = b, \tag{1}
\]
where \( H : \mathbb{R}^n \rightarrow \mathbb{R} \) and \( G : \mathbb{R}^m \rightarrow \mathbb{R} \) 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
\[
\begin{align*}
    u_{k+1} &= \arg \min_u H(u) + \langle \lambda_k, -Au \rangle + \frac{\tau_k}{2} \| b - Au - Bv_k \|^2_2 \\
    v_{k+1} &= \arg \min_v G(v) + \langle \lambda_k, -Bv \rangle + \frac{\tau_k}{2} \| b - Au - Bv \|^2_2 \\
    \lambda_{k+1} &= \lambda_k + \tau_k (b - Au_{k+1} - Bv_{k+1}),
\end{align*}
\]
where the sequence of penalties \( \tau_k \) is the only free choice, and has a high impact on the algorithm’s performance. Our goal is to automate this choice, by adaptively tuning \( \tau_k \) 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
\[
r_k = b - Au_k - Bv_k, \quad d_k = \tau_k A^T B (v_k - v_{k-1}),
\]
respectively \([2]\). The iteration is generally stopped when \( \|r_k\|_2 \leq \epsilon_{tol} \max \{\|Au_k\|_2, \|Bv_k\|_2, \|b\|_2 \} \) and \( \|d_k\|_2 \leq \epsilon_{tol} \|A^T \lambda_k\|_2 \), where \( \epsilon_{tol} > 0 \) is the stopping tolerance (e.g., \( \epsilon_{tol} \approx 10^{-3} \)).

### 2.2 Parameter tuning and adaptation

Relatively little work has been done on automating ADMM, i.e., on adaptively choosing \( \tau_k \). In the specific case where the objective is a strictly convex quadratic function, criteria for choosing an optimal constant penalty have been recently proposed \([2] [19]\).

Residual balancing (RB \([2] [15]\)) is the only available adaptive method for general form problems \([1]\). RB is based on the following observation: increasing \( \tau_k \) strengthens the penalty term, resulting in smaller primal residuals but larger dual ones; conversely, decreasing \( \tau_k \) leads to larger primal and smaller dual residuals. Since both residuals must be small at convergence, it makes sense to “balance” them, i.e., to tune \( \tau_k \) to keep both residuals of similar magnitude. A simple scheme for this goal is
\[
\tau_{k+1} = \begin{cases} 
    \eta \tau_k & \text{if } \|d_k\|_2 > \mu \|r_k\|_2 \\
    \tau_k / \eta & \text{if } \|r_k\|_2 > \mu \|d_k\|_2 \\
    \tau_k & \text{otherwise},
\end{cases}
\]
for parameters \( \mu > 1 \) and \( \eta > 1 \) \([2]\). RB has recently been adapted to distributed optimization \([22]\) and other primal-dual splitting methods \([13]\). ADMM with adaptive penalty is not guaranteed to converge, although convergence can be enforced by fixing \( \tau_k = \tau \) after a finite number of iterations \([15]\).

The RB idea suffers from a fundamental flaw. The relative size of the residuals depends on the (arbitrary) scaling of the problem; for example, with the change of variable \( u \leftarrow 10 u \), 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 for some problems, 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 \( \eta \) and \( \mu \), the penalty parameters may oscillate and the algorithm fails to converge before \( \tau_k \) is fixed.

### 2.3 Dual interpretation of ADMM

We now explain the close relationship between ADMM and Douglas-Rachford Splitting (DRS) \([4] [6] [11]\), which plays a central role in the proposed approach. The starting observation is that the dual of problem \([1]\) has the form
\[
\min_{\zeta \in \mathbb{R}^p} \underbrace{H^*(A^T \zeta) - \langle \zeta, b \rangle + G^*(B^T \zeta)}_{H(\zeta)} + \langle \hat{H}(\zeta), \hat{G}(\zeta) \rangle,
\]
with \( F^* \) denoting the Fenchel conjugate of \( F \), defined as \( F^*(y) = \sup_x \langle x, y \rangle - F(x) \) \([20]\).

The DRS algorithm solves \((7)\) by generating two sequences \((\zeta_k)_{k \in \mathbb{N}}\) and \((\hat{\zeta}_k)_{k \in \mathbb{N}}\) according to
\[
\begin{align*}
    0 &\in \frac{\zeta_{k+1} - \zeta_k}{\tau_k} + \partial \hat{H}(\hat{\zeta}_{k+1}) + \partial \hat{G}(\zeta_k) \\
    0 &\in \frac{\hat{\zeta}_{k+1} - \hat{\zeta}_k}{\tau_k} + \partial \hat{H}(\hat{\zeta}_{k+1}) + \partial \hat{G}(\zeta_{k+1}),
\end{align*}
\]
where we use the standard notation $\partial F(x)$ for the subdifferential of $F$ evaluated at $x$.

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 of (2) is

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

which is equivalent to $A^T \hat{\lambda}_{k+1} \in \partial H(u_{k+1})$, thus

$$u_{k+1} \in \partial H^* (A^T \hat{\lambda}_{k+1}).$$

A similar argument using the optimality condition for (3) leads to $u_{k+1} \in \partial G^* (B^T \hat{\lambda}_{k+1})$. Recalling (7), we arrive at

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

Using these identities, we finally have

$$\hat{\lambda}_{k+1} = \lambda_k + \tau_k (b - Au_{k+1} - Bv_k) \in \lambda_k - \tau_k (\partial \tilde{H}(\hat{\lambda}_{k+1}) + \partial \tilde{G}(\lambda_k))$$

and

$$\lambda_{k+1} = \lambda_k + \tau_k (b - Au_{k+1} - Bv_{k+1}) \in \lambda_k - \tau_k (\partial \tilde{H}(\hat{\lambda}_{k+1}) + \partial \tilde{G}(\lambda_{k+1})).$$

showing that the sequences $(\lambda_k)_{k \in \mathbb{N}}$ and $(\hat{\lambda}_k)_{k \in \mathbb{N}}$ satisfy the same conditions (8) and (9) 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 (7).

### 2.4 Spectral (Barzilai-Borwein) stepsize selection

The classical gradient descent step for unconstrained minimization of a smooth function $F: \mathbb{R}^n \to \mathbb{R}$ has the form $x_{k+1} = x_k - \tau_k \nabla F(x_k)$. Spectral gradient methods, pioneered by Barzilai and Borwein (BB) [1], adaptively choose the stepsize $\tau_k$ to achieve fast convergence. In a nutshell, the standard (there are variants) BB method sets $\tau_k = 1/\alpha_k$, with $\alpha_k$ chosen such that $\alpha_k I$ mimics the Hessian of $F$ over the last step, seeking a quasi-Newton step. Using a least squares criterion yields

$$\alpha_k = \arg\min_{\alpha \in \mathbb{R}} \| \nabla F(x_k) - \nabla F(x_{k-1}) - \alpha (x_k - x_{k-1}) \|^2_2,$$

which is an estimate of the curvature of $F$ across the previous step of the algorithm. Spectral gradient methods dramatically outperform schemas with constant stepsize in many applications [7, 24] and have been generalized to handle non-differentiable problems via proximal gradient methods [23, 12].

Finally, notice that (14) is equivalent to modeling the gradient $\nabla F(x_k)$ as a linear function of $x_k$,

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

where $a_k = \nabla F(x_{k-1}) - \alpha_k x_{k-1} \in \mathbb{R}^n$. 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 [1], 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 methods to prevent unexpected behavior when curvature estimates are inaccurate.

#### 3.1 Spectral stepsize for Douglas-Rachford splitting

Considering the dual problem (7), and following the observation in (15) about the BB method, we model/approximate $\partial \tilde{H}(\zeta)$ and $\partial \tilde{G}(\zeta)$ at iteration $k$ as linear functions of their arguments,

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

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

**Proposition 1 (Spectral DRS).** Suppose the DRS steps (3)–(9) are applied to problem (7), where (omitting the subscript $k$ from $\alpha_k, \beta_k, \Psi_k, \Phi_k$ to lighten the notation in what follows)

$$\partial \tilde{H}(\zeta) = \alpha \dot{\zeta} + \Psi \quad \text{and} \quad \partial \tilde{G}(\zeta) = \beta \dot{\zeta} + \Phi.$$

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

\(^1\)An important property relating $F$ and $F^*$ is that $y \in \partial H(x)$ if and only if $x \in \partial H^*(y)$ [20].
Proof. Inserting (16) into the DRS step (8)–(9), we have

\[
0 \equiv \frac{\hat{\zeta}_{k+1} - \zeta_k}{\tau} + (\alpha \hat{\zeta}_{k+1} + \Psi) + (\beta \zeta_k + \Phi)
\]  

(17)

\[
0 \equiv \frac{\hat{\zeta}_{k+1} - \zeta_k}{\tau} + (\alpha \hat{\zeta}_{k+1} + \Psi) + (\beta \zeta_{k+1} + \Phi).
\]  

(18)

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

\[
\hat{\zeta}_{k+1} = \frac{1 - \beta \tau}{1 + \alpha \tau} \zeta_k - \frac{a \tau + b \tau}{1 + \alpha \tau},
\]  

(19)

where \( a \in \Psi \) and \( b \in \Phi \), and for \( \zeta_{k+1} \) as

\[
\zeta_{k+1} = \frac{1}{1 + \beta \tau} \zeta_k - \frac{a \tau + b \tau}{1 + \beta \tau} \hat{\zeta}_{k+1} = \frac{1 + \alpha \beta \tau^2}{(1 + \alpha \tau)(1 + \beta \tau)} (\hat{\zeta}_k - (a + b) \tau),
\]  

(20)

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

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

\[
r_{DR} = \| (\alpha + \beta) \zeta_{k+1} + (a + b) \| = \frac{1 + \alpha \beta \tau^2}{(1 + \alpha \tau)(1 + \beta \tau)} \| (\alpha + \beta) \zeta_k + (a + b) \|,
\]  

(21)

where \( \zeta_{k+1} \) in (21) was substituted with (20). The optimal stepsize \( \tau_k \) minimizes the residual

\[
\tau_k = \arg \min_\tau r_{DR} = \arg \max_\tau \frac{(1 + \alpha \tau)(1 + \beta \tau)}{1 + \alpha \beta \tau^2} = \arg \max_\tau \frac{(\alpha + \beta) \tau}{1 + \alpha \beta \tau^2} = 1/\sqrt{\alpha \beta}.
\]  

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 \( \zeta_k \) and \( \zeta_{k+1} \), respectively.

\[\square\]

### 3.2 Spectral stepsize estimation

Proposition 1 shows how to adaptively choose the penalty parameters. We can begin by obtaining linear estimates of the subgradients of the terms in the dual objective (7). The geometric mean of the optimal penalty parameter for ADMM, thanks to the equivalence presented in Subsection 2.3.

We now address the question of how to estimate \( \hat{\alpha}_k = \alpha_k^{-1} \) and \( \hat{\beta}_k = \beta_k^{-1} \) for the components \( \hat{G}(\hat{\lambda}_k) \) and \( \hat{H}(\hat{\lambda}_k) \) at the \( k \)-th iteration. The curvature parameters are estimated based on the results from iteration \( k \) and an older iteration \( k_0 < k \). Noting the identities (11), we define

\[
\Delta \hat{\lambda}_k := \hat{\lambda}_k - \hat{\lambda}_{k_0} \quad \text{and} \quad \Delta \hat{H}_k := \partial \hat{H}(\hat{\lambda}_k) - \partial \hat{H}(\hat{\lambda}_{k_0}) = A(u_k - u_{k_0}).
\]  

(22)

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 the spectral stepsize literature [24], the curvature of \( \hat{H}(\hat{\lambda}) \) is estimated via one of the two least squares problems

\[
\min_\alpha \| \Delta \hat{H}_k - \alpha \Delta \hat{\lambda}_k \|^2_2 \quad \text{or} \quad \min_\alpha \| \alpha^{-1} \Delta \hat{H}_k - \Delta \hat{\lambda}_k \|^2_2.
\]  

(23)

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

\[
\hat{\alpha}^SD_k = \frac{\langle \Delta \hat{\lambda}_k, \Delta \hat{\lambda}_k \rangle}{\langle \Delta \hat{H}_k, \Delta \hat{\lambda}_k \rangle} \quad \text{and} \quad \hat{\alpha}^MG_k = \frac{\langle \Delta \hat{H}_k, \Delta \hat{\lambda}_k \rangle}{\langle \Delta \hat{H}_k, \Delta \hat{H}_k \rangle},
\]  

(24)

where (following [24]) SD stands for steepest descent stepsize, and MG for minimum gradient stepsize. The Cauchy-Schwarz inequality can be used to show that \( \hat{\alpha}^SD_k \geq \hat{\alpha}^MG_k \). Rather than choosing one or the other, we suggest the hybrid stepsize rule proposed in [12, 24], defined as

\[
\hat{\alpha}_k = \begin{cases} 
\hat{\alpha}^MG_k & \text{if } 2 \hat{\alpha}^MG_k > \hat{\alpha}^SD_k \\
\hat{\alpha}^SD_k - \hat{\alpha}^MG_k / 2 & \text{otherwise.}
\end{cases}
\]  

(25)
The spectral stepsize $\hat{\beta}_k = 1/\beta_k$ of $\hat{G}(\lambda_k)$ is similarly estimated as,

$$
\hat{\beta}_k = \begin{cases} 
\hat{\beta}^{\text{MG}}_k & \text{if } \hat{\beta}^{\text{MG}}_k > \hat{\beta}^{\text{SD}}_k \\
\hat{\beta}^{\text{SD}}_k - \hat{\beta}^{\text{MG}}_k / 2 & \text{otherwise},
\end{cases}
$$

where $\hat{\beta}^{\text{SD}}_k = \langle \Delta \lambda_k, \Delta \lambda_k \rangle / \langle \Delta \hat{G}_k, \Delta \lambda_k \rangle$, $\hat{\beta}^{\text{MG}}_k = \langle \Delta \hat{G}_k, \Delta \lambda_k \rangle / \langle \Delta \hat{G}_k, \Delta \hat{G}_k \rangle$, $\Delta \hat{G}_k = B(v_k - v_{k_0})$, 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 alone, i.e., our scheme does not require the user to supply the dual problem.

### 3.3 Safeguarding: testing the quality of stepsize estimates

On some iterations, the linear models for one or both subgradients that underly the spectral stepsize choice may be very inaccurate. When this occurs, the least squares procedure may produce ineffective stepsize values. The classical BB method for unconstrained problems uses a line search to safeguard against unstable stepizes when curvature estimates are unreliable. For 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 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):

$$
\alpha_k^{\text{corr}} = \frac{\langle \Delta \hat{H}_k, \Delta \hat{\lambda}_k \rangle}{\| \Delta \hat{H}_k \| \| \Delta \hat{\lambda}_k \|} \quad \text{and} \quad \beta_k^{\text{corr}} = \frac{\langle \Delta \hat{G}_k, \Delta \lambda_k \rangle}{\| \Delta \hat{G}_k \| \| \Delta \lambda_k \|}
$$

The correlations indicate whether the linear assumptions are suitable to estimate the gradients, thus the spectral stepsizes $\hat{\alpha}_k$ and $\hat{\beta}_k$ are utilized only if the correlations indicate the estimation is credible enough. Finally, the safeguarded spectral adaptive penalty parameter rule is

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

where $\epsilon^{\text{corr}}$ is a quality threshold for the curvature estimates, while $\hat{\alpha}_k$ and $\hat{\beta}_k$ are the spectral stepsizes given by (25) and (26). The proposed method falls back to constant penalty parameter when both curvature estimates are deemed inaccurate.

### 3.4 Adaptive ADMM with spectral penalty parameter

The complete adaptive ADMM (AADMM) is shown in Algorithm 1. We suggest only updating the stepsize every $T_f$ iterations. The overhead of the proposed adaptivity scheme is modest, requiring only a few inner products, plus the storage needed to hold one previous iterate. As noted in [15], convergence is guaranteed if the adaptivity is turned off after a finite number of iterations; however, we have found this to be unnecessary in practice.

### 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 [11], quadratic programming (QP) [2, 9, 11, 19], basis pursuit [2, 11], and consensus $\ell_1$-regularized logistic regression [2]. We use both synthetic and benchmark datasets (obtained from the UCI repository and the LIBSVM page) used in [5, 16, 17, 21, 23, 25].

For comparison, we implemented vanilla ADMM, fast ADMM with a restart strategy [11], and ADMM with residual balancing [2, 13], using $\mu = 10$ and $\eta = 2$ in [9], and turned off after 1000 iterations.
Algorithm 1 Adaptive ADMM (AADMM) with spectral penalty parameter selection

**Input:** initialize \( v_0, \lambda_0, \tau_0, k_0 = 0, e^{\text{corr}} = 0.2, \) update frequency \( T_f = 2 \)

1: for \( k = 0, 1, 2, \ldots, \) maxiter do
2: \( u_{k+1} \leftarrow \arg\min_u H(u) + \langle \lambda_k, -Au \rangle + \frac{\rho}{2} \| b - Au - Bv_k \|^2 \)
3: \( v_{k+1} \leftarrow \arg\min_v G(v) + \langle \lambda_k, -Bv \rangle + \frac{\rho}{2} \| b - Au_{k+1} - Bv \|^2 \)
4: \( \lambda_{k+1} = \lambda_k + \tau_k (b - Au_{k+1} - Bv_{k+1}) \)
5: if \( \text{mod}(k, T_f) = 1 \) then
6: \( \lambda_{k+1} = \lambda_k + \tau_k (b - Au_{k+1} - Bv_{k+1}) \)
7: Compute spectral stepsizes \( \hat{\alpha}_k, \hat{\beta}_k \) using (25) and (26)
8: Estimate correlations \( \hat{\alpha}^{corr}_k, \hat{\beta}^{corr}_k \), as given in (27)
9: end if
10: for \( k_0 \leftarrow k \) do
11: end if
12: \( \tau_{k+1} \leftarrow \tau_k \)
13: end for

iterations to guarantee convergence. The proposed AADMM is implemented as shown in Algorithm 1 with fixed parameters \( e^{\text{corr}} = 0.2 \) and \( T_f = 2 \).

We set the stopping tolerance to \( e^{\text{tol}} = 10^{-5}, 10^{-3}, \) and 0.05 for small, medium, and large scale problems, respectively. The initial penalty \( \tau_0 = 1/10 \) is used for all problems except the general QP problem, where \( \tau_0 \) is set to the value proposed for quadratic problems in [19].

4.2 Applications

**Linear regression with elastic net (EN) regularization.** EN is a modification of the \( \ell_1 \) (or LASSO) regularizer that helps preserving groups of highly correlated variables [11, 25], and requires solving

\[
\min_x \frac{1}{2} \| Dx - c \|^2 + \rho_1 \| x \|_1 + \frac{\rho_2}{2} \| x \|^2_2,
\]

where \( \| \cdot \|_1 \) denotes the \( \ell_1 \) norm, \( D \) is a data matrix, \( c \) contains measurements, and \( x \) is the regression coefficients. One way to apply ADMM to this problem is to rewrite it as

\[
\min_{u,v} \frac{1}{2} \| Du - c \|^2 + \rho_1 \| v \|_1 + \frac{\rho_2}{2} \| v \|^2_2 \quad \text{subject to} \quad u - v = 0.
\]

The synthetic dataset introduced in [11, 25] and realistic dataset introduced in [5, 25] are investigated. Typical parameters \( \rho_1 = \rho_2 = 1 \) are used in all experiments.

**Support vector machine (SVM) and QP.** The dual of the SVM learning problem is a QP

\[
\min_{z} \frac{1}{2} z^T Q z - c^T z \quad \text{subject to} \quad c^T z = 0 \quad \text{and} \quad 0 \leq z \leq C,
\]

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 \( C > 0 \) [3]. We also consider the canonical QP

\[
\min_x \frac{1}{2} x^T Q x + q^T x \quad \text{subject to} \quad D x \leq c,
\]

which could be solved by applying ADMM to

\[
\min_{u,v} \frac{1}{2} u^T Q u + q^T u + \iota_{\{z: z_i \leq c\}}(v) \quad \text{subject to} \quad Du - v = 0;
\]

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

We study classification problems from [16, 21] with \( C = 1 \) and a random synthetic QP from [11], where \( Q \in \mathbb{R}^{500 \times 500} \) with condition number approximately \( 4.5 \times 10^5 \).

**Basis pursuit (BP) and sparse representation.** BP solves the constrained minimization problem

\[
\min_x \| x \|_1 \quad \text{subject to} \quad D x = c,
\]

(34)
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 \[24\]. To apply ADMM, problem \([34]\) is rewritten as

\[
\min_{u,v} \ell(z; Dz = c)(u) + \|v\|_1 \quad \text{subject to} \quad u - v = 0.
\]  

(35)

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 \[23\], where each frontal face image is scaled to \( 32 \times 32 \). 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 \( \ell_1 \)-regularized logistic regression.** ADMM has become an important tool for solving distributed problems \[2\]. Here, we consider the consensus \( \ell_1 \)-regularized logistic regression

\[
\min_{x,z} \sum_{i=1}^{N} n_i \sum_{j=1}^{m} \log(1 + \exp(-c_j D_j x_i)) + \rho \|z\|_1 \quad \text{subject to} \quad x_i - z = 0, i = 1, \ldots, N.
\]  

(36)

where \( x_i \in \mathbb{R}^m \) represents the local variable on the \( i \)th distributed node, \( z \) is the global variable, \( n_i \) is the number of samples in the \( i \)th block, \( D_j \in \mathbb{R}^{m} \) is the \( j \)th sample, and \( c_j \in [-1, 1] \) is the corresponding label. The synthetic problem is constructed with Gaussian random data and sparse ground truth solutions. Binary classification problems from \[16, 17, 21\] 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.

### 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 9 out of 17 realistic datasets, it fails to converge in the maximum number of iterations \[9\]. Fast ADMM \[11\] 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.

| Application | Dataset   | #samples × #features | Vanilla ADMM | Fast ADMM          | Residual balance | Adaptive ADMM |
|-------------|-----------|----------------------|--------------|--------------------|------------------|---------------|
|             |           |                      |              |                    |                  |               |
| Elastic net | Synthetic | 50 × 40              | 2000+ (1.64) | 263 (2.70)         | 111 (1.29)       | 43 (0.46)     |
| regression  | Boston    | 506 × 13             | 2000+ (2.19) | 208 (1.06)         | 54 (0.023)       | 17 (0.11)     |
|             | Diabetes  | 768 × 8              | 974 (2.69)   | 947 (8.48)         | 28 (0.020)       | 10 (0.005)    |
|             | Leukemia  | 38 × 7129            | 2000+ (22.9) | 2000+ (24.2)       | 1737 (19.3)      | 152 (1.70)    |
|             | Prostate  | 97 × 8               | 548 (2.93)   | 139 (0.49)         | 29 (0.015)       | 16 (0.012)    |
|             | Servo     | 130 × 4              | 142 (0.40)   | 44 (0.017)         | 27 (0.012)       | 13 (0.007)    |
| QP and dual | Synthetic | 250 × 200            | 439 (6.15)   | 535 (7.8380)       | 232 (3.27)       | 71 (0.984)    |
| SVM         | Madelon   | 2000 × 500           | 100 (14.0)   | 57 (8.14)          | 28 (4.12)        | 19 (2.64)     |
|             | Sonar     | 208 × 60             | 139 (2.27)   | 43 (0.75)          | 37 (0.069)       | 28 (0.050)    |
|             | Splice    | 1000 × 60            | 149 (4.9)    | 47 (1.44)          | 39 (1.27)        | 20 (0.681)    |
| Basis pursuit | Synthetic | 10 × 30              | 163 (0.27)   | 2000+ (3.10)       | 159 (0.31)       | 114 (0.026)   |
|             | Human1    | 1024 × 1087          | 2000+ (2.35) | 2000+ (2.41)       | 839 (0.990)      | 503 (0.626)   |
|             | Human2    | 1024 × 1087          | 2000+ (2.26) | 2000+ (2.42)       | 875 (1.03)       | 448 (0.554)   |
|             | Human3    | 1024 × 1087          | 2000+ (2.29) | 2000+ (2.44)       | 713 (8.55)       | 523 (0.641)   |
| Consensus logistic regression | Synthetic | 1000 × 25            | 301 (3.36)   | 444 (3.54)         | 43 (3.583)       | 27 (0.282)    |
|             | Madelon   | 2000 × 500           | 2000+ (205)  | 2000+ (166)        | 115 (42.1)       | 23 (0.08)     |
|             | 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 (0.638)    |
|             | 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)      |

\(^{2}2000 \text{ for small and medium datasets and 1000 for large datasets.}\)

\(^{3}\text{#constrains } \times \text{ #unknowns for general QP and basis pursuit}\)
4.4 Sensitivity to initial stepsize and problem scaling

Finally, we study the sensitivity of the different ADMM variants to the initial penalty parameter ($\tau_0$) choice and problem scaling. Fig. 1 presents iteration counts for a wide range of values of $\tau_0$, for elastic net regression (left) and general QP (right) with synthetic datasets. Scaling sensitivity experiments were done by multiplying the measurement vector $c$ in elastic net and QP by a scalar $s$. Fast ADMM and vanilla ADMM use the fixed initial penalty parameter $\tau_0$, and are highly sensitivity to this choice, as shown in Fig. 1; in contrast, AADMM is stable with respect to the initial $\tau_0$ and the problem scale $s$.

![Figure 1: Sensitivity of convergence speed to the initial penalty parameter $\tau_0$ for EN (left) and QP (right).](image1)

![Figure 2: Sensitivity of convergence speed to data scaling for the synthetic problems of EN (left) and QP (right).](image2)

5 Conclusion

We have proposed adaptive ADMM (AADMM), a new variant of the very 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 wide range of problems and datasets have shown that AADMM outperforms other variants of ADMM. AADMM was also shown to be robust with respect to initial parameter choice and problem scaling.
References

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

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

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

[4] 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.

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

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

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

[8] 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.

[9] 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.

[10] R. Glowinski and A. Marroco. Sur l’approximation, par éléments finis d’ordre un, et la résolution, par pénalisation-dualité d’une classe de problèmes de Dirichlet non linéaires. *ESAIM: Modélisation Mathématique et Analyse Numérique*, 9:41–76, 1975.

[11] T. Goldstein, B. O’Donoghue, S. Setzer, and R. Baraniuk. Fast alternating direction optimization methods. *SIAM Journal on Imaging Sciences*, 7(3):1588–1623, 2014.

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

[13] 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.

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

[15] 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.

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

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

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

[19] 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.

[20] R. Rockafellar. *Convex Analysis*. Princeton University Press, 1970.

[21] 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.

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

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

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

[25] 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.