Implementing the ADMM to Big Datasets: A Case Study of LASSO

Hangrui Yue* Qingzhi Yang† Xiangfeng Wang‡ Xiaoming Yuan§

August 1, 2017

Abstract

The alternating direction method of multipliers (ADMM) has been popularly used for a wide range of applications in the literature. When big datasets with high-dimensional variables are considered, subproblems arising from the ADMM must be solved inexactly even though theoretically they may have closed-form solutions. Such a scenario immediately poses mathematical ambiguities such as how accurately these subproblems should be solved and whether or not the convergence can be still guaranteed. Despite of the popularity of ADMM, it seems not too much is known in these regards. In this paper, we look into the mathematical detail of implementing the ADMM to such big-data scenarios. More specifically, we focus on the convex programming case where there is a quadratic function component with extremely high-dimensional variables in the objective of the model under discussion and thus there is a huge-scale system of linear equations to be solved at each iteration of the ADMM. We show that there is no need (indeed it is impossible) to solve this linear system exactly or too accurately; and propose an automatically adjustable inexactness criterion to solve these linear systems inexactly. We further identify the safe-guard numbers for the internally nested iterations that can sufficiently ensure this inexactness criterion if these linear systems are solved by standard numerical linear algebra solvers. The convergence, together with worst-case convergence rate measured by the iteration complexity, is rigorously established for the ADMM with inexactly-solved subproblems. Some numerical experiments for big datasets of the LASSO with millions of variables are reported to show the efficiency of this inexact implementation of ADMM.

Keywords: Convex programming; Alternating direction method of multipliers; High dimension; Big data; LASSO; Distributed LASSO; Convergence

1 Introduction

We discuss how to implement the alternating direction method of multipliers (ADMM) to big datasets in the convex programming context, and present the rigorous mathematical analysis for its convergence. The ADMM was originally proposed in [6, 19] for nonlinear elliptic equations and it recently

*School of Mathematical Sciences and LPMC, Nankai University, Tianjin, P.R. China. This author was supported by the NSFC, Grant No. 11671217 and by the Ph.D. Candidate Research Innovation Foundation of Nankai University. Email: yuehangrui@gmail.com.

†School of Mathematical Sciences and LPMC, Nankai University, Tianjin, P.R. China. This author was supported by the NSFC, Grants No. 11271206 and 11671217. Email: qz-yang@nankai.edu.cn.

‡Shanghai Key Lab for Trustworthy Computing, School of Computer Science and Software Engineering, East China Normal University, Shanghai, P. R. China. This author was supported by NSFC, Grant No.11501210. Email: xfwang@sei.ecnu.edu.cn.

§Department of Mathematics, Hong Kong Baptist University, Hong Kong, P. R. China. This author was supported by a General Research Fund from Hong Kong Research Grants Council. Email: xmyuan@hkbu.edu.hk.

has found a wide range of applications in various areas such as image processing, statistical learning, computer vision, wireless communication network, and so on. It becomes a benchmark first-order solver for convex minimization models with separable objective functions, and is being extensively explored in other various contexts such as the nonconvex or multi-block contexts. We refer to [5, 11, 18] for some review papers of the ADMM.

Let us focus on the separable convex programming problem with linear constraints and an objective function in form of two functions without coupled variables:

$$\begin{aligned} \min_{x\in\mathbb{R}^n, y\in\mathbb{R}^m} \quad & \underbrace{\frac{1}{2}\|Qx-q\|_2^2}_{f(x)} + g(y) \\ \text{s.t.} \quad & Ax+By=b, \end{aligned} \tag{1.1}$$

where $Q\in\mathbb{R}^{p\times n}$; $q\in\mathbb{R}^p$; $g:\mathbb{R}^m\to\mathbb{R}$ is a general convex (not necessarily smooth) closed function; $A\in\mathbb{R}^{\ell\times n}$; $B\in\mathbb{R}^{\ell\times m}$ and $b\in\mathbb{R}^\ell$. Instead of considering a generic convex function $f(x)$ in (1.1), we just focus on the quadratic case because our emphasis, as to be delineated, is the implementation of ADMM when the x-subproblem at each iteration is a system of linear equations. We particularly assume $p \ll n$ so that the model (1.1) captures the fundamental model of least absolute shrinkage and selection operator (LASSO):

$$\min_{x\in\mathbb{R}^n} \ \frac{1}{2}\|Qx-q\|_2^2 + \tau\|x\|_1. \tag{1.2}$$

Note that the LASSO model (1.2) can be easily reformulated as the special case of (1.1) with $A=I_{n\times n}$, $B=-I_{n\times n}$, $b=0$, $g(y)=\tau\|y\|_1$; see (1.6) for details. For the LASSO problem (1.2), the matrix Q is a data fidelity term and the term $\|x\|_1$ prompts the sparsest solution of the underdetermined system of linear equations $Qx=q$ with $p\ll n$. The LASSO model (1.2) was initiated in [39], and it is fundamental in several fields such as compressive sensing [9], statistical learning [15], MRI medical image processing [29], radar signal recovery [3], robust feature selection in machine learning [34], and etc. More sophisticated applications of the LASSO model (1.2) also include various distributed optimization problems arising in multi-agent network models such as those in [1, 4, 7, 8, 33, 44]. Our analysis will be applicable to the more general model (1.1), but with an exclusive emphasis on the LASSO model (1.2) because of its importance. Throughout, we assume that the matrix A in (1.1) is full column rank and the inverse of A^TA can be computed easily; and the solution set of (1.1) is nonempty to avoid triviality.

Let the augmented Lagrangian function of (1.1) be defined as

$$\mathcal{L}_\beta(x,y,\lambda) = f(x)+g(y)-\lambda^T(Ax+By-b)+\frac{\beta}{2}\|Ax+By-b\|_2^2, \tag{1.3}$$

with $\lambda\in\mathbb{R}^\ell$ the Lagrange multiplier and $\beta>0$ a penalty parameter. Then, the iterative scheme of ADMM for (1.1) reads as

$$\begin{cases} x^{k+1} := \arg\min_{x\in\mathbb{R}^n} \mathcal{L}_\beta\left(x, y^k, \lambda^k\right), & \text{(1.4a)} \\ y^{k+1} := \arg\min_{y\in\mathbb{R}^m} \mathcal{L}_\beta\left(x^{k+1}, y, \lambda^k\right), & \text{(1.4b)} \\ \lambda^{k+1} := \lambda^k - \beta\left(Ax^{k+1}+By^{k+1}-b\right). & \text{(1.4c)} \end{cases}$$

It is easy to see that the subproblems in the ADMM scheme (1.4) are generally easier than the original problem (1.1). Indeed, it is commented in [5] that "*a simple algorithm derived from basic*

ADMM will often offer performance that is at least comparable to very specialized algorithms (even in the serial setting), and in most cases, the simple ADMM algorithm will be efficient enough to be useful.". To simplify the discussion, we assume that β is fixed in our theoretical analysis even though it does need to discuss how to adjust it dynamically for numerical implementation purpose.

It worths to mention that for some special cases, (1.4a) and (1.4b) may be easy enough to have closed-form solutions and thus no internal iterations are involved when implementing (1.4); see, e.g., (1.9) and (1.10) when the LASSO model (1.2) is considered. This feature indeed is the main reason that the ADMM finds many efficient applications in the mentioned areas. Generically, (1.4a) and (1.4b) should be solved iteratively and only approximate solutions can be pursued by internal iterations. Hence, generally the ADMM (1.4) should be implemented as the following with internally nested iterations:

$$\begin{cases} x^{k+1} \approx \arg\min\limits_{x\in\mathbb{R}^n} \mathcal{L}_\beta\left(x, y^k, \lambda^k\right), \\ y^{k+1} \approx \arg\min\limits_{y\in\mathbb{R}^m} \mathcal{L}_\beta\left(x^{k+1}, y, \lambda^k\right), \\ \lambda^{k+1} := \lambda^k - \beta\left(Ax^{k+1} + By^{k+1} - b\right). \end{cases} \tag{1.5}$$

It is mathematically important to well define the inexactness criterion in (1.5) and study the rigorous convergence with the x- and y-subproblems solved inexactly. It should be mentioned that although the convergence of ADMM has been well studied in both earlier literature (e.g., [10, 14, 16, 17, 20, 23]) and recent papers [24, 25, 32], these results are valid only for the exact version (1.4) in which both (1.4a) and (1.4b) are assumed to be solved exactly. Hence, the convergence of (1.5) with internally nested iterations should be analyzed from scratch. When the generic case (1.5) is considered, a general rule of guaranteeing the convergence of (1.5) is that the accuracy of an inexactness criterion should keep increasing as iterations go on. While how to efficiently specify an inexactness criterion and the accuracy in (1.5), which is an important issue for numerical implementation, can be discussed in a specific scenario of the generic model (1.1). In this paper, we focus on the latter case and refer to [10, 11, 12, 22, 35] for the former case.

There are different choices of algorithms for solving the LASSO model (1.2); and the ADMM (1.4) is a competitive one, see, e.g., [5]. Let us delineate the detail of the application of ADMM (1.4) to the LASSO model (1.2), and use this application to illustrate our idea of dealing with the mathematical issues arising from implementing the ADMM to some big-data scenarios. First, introducing an auxiliary variable $y \in \mathbb{R}^n$, we explicitly rewrite the LASSO model (1.2) in form of (1.1):

$$\min_{x\in\mathbb{R}^n, y\in\mathbb{R}^n} \quad \frac{1}{2}\|Qx - q\|_2^2 + \tau\|y\|_1 \qquad \text{s.t.} \qquad x = y, \tag{1.6}$$

and the ADMM scheme (1.4) can be accordingly specified as

$$\begin{cases} x^{k+1} = \arg\min_{x\in\mathbb{R}^n}\left\{\frac{1}{2}\|Qx - q\|_2^2 + \frac{\beta}{2}\left\|x - y^k - \frac{\lambda^k}{\beta}\right\|_2^2\right\}, \\ y^{k+1} = \arg\min_{y\in\mathbb{R}^n}\left\{\tau\|y\|_1 + \frac{\beta}{2}\left\|y - x^{k+1} + \frac{\lambda^k}{\beta}\right\|_2^2\right\}, \\ \lambda^{k+1} = \lambda^k - \beta\left(x^{k+1} - y^{k+1}\right). \end{cases} \tag{1.7}$$

Methodologically, the implementation of (1.7) seems to be easy. The x-subproblem in (1.7) can be expressed as

$$\left(Q^TQ + \beta I\right)x = Q^Tq + \beta\left(y^k + \frac{\lambda^k}{\beta}\right) \tag{1.8}$$

and its solution is analytically given by

$$x^{k+1} = \left(Q^T Q + \beta I\right)^{-1} \left[Q^T q + \beta \left(y^k + \frac{\lambda^k}{\beta}\right)\right]; \tag{1.9}$$

while the solution of the y-subproblem is given by

$$y^{k+1} = \mathcal{S}\left(x^{k+1} - \frac{\lambda^k}{\beta}, \frac{\tau}{\beta}\right) \tag{1.10}$$

where $\mathcal{S}(x, a)$ denotes the shrinkage operator (see [39]), *i.e.*, $\mathcal{S}(x, a) := x - \max\{\min\{x, a\}, -a\}$.

Despite of the closed-form solution given in (1.9), we consider the big-data scenario of the LASSO model (1.2) with high-dimensional variables, *i.e.*, both p and n could be huge. For such a big-data scenario, it becomes impossible or extremely expensive to solve the linear system (1.8) exactly, by neither direct nor iterative methods. Indeed, even though standard numerical linear algebra solvers such as the conjugate gradient (CG) or preconditioned conjugate gradient (PCG) methods guarantee finding the solution of (1.8) after n steps, it should not be considered to execute all the n iterations when n is huge. We thus need to solve the system (1.8) iteratively at each iteration. Some questions arise immediately: how accurate should an iterate be if a specific solver is applied to solve the linear system (1.8); and for implementation purpose, how many iterations should be implemented for solving the linear system (1.8)? For example, according to the authors' website[1], see also [5, Section 8.2.1], it is suggested to solve the linear system (1.8) by the LSQR solver (see [36]) with the fixed accuracy of 10^{-6}. It becomes interesting to ask if the convergence of ADMM can be still guaranteed when all the x-subproblems are solved inexactly with a fixed accuracy. On the other hand, it seems to be puzzled to fix which level of accuracy a prior, because neither too high nor too low accuracy is good for generating satisfactory numerical performance. Indeed, if the accuracy is fixed, then it needs to tune the level of accuracy a prior and the "optimal" level of accuracy may vary from different specific applications of the generic model (1.1). Also, there is no obvious justification to testify that pursuing an exact solution of the x-subproblem, or an approximate solution with a high accuracy, is necessary at the earlier stage of the iteration. All these questions urge us to consider finding an inexactness criterion that can adjust the accuracy dynamically and automatically so as to solve the linear system (1.8) inexactly for the big-data scenario of the LASSO model (1.2), and to rigorously prove the convergence of the inexact version of ADMM with this inexactness criterion.

Our main findings are: (1) proposing an automatically adjustable inexactness criterion for solving (1.4a) inexactly and thus proposing an inexact version of the ADMM for the model (1.1); (2) rigorously proving the convergence of the inexact version of ADMM and estimating its convergence rate in terms of the iteration complexity; and (3) specifying the safe-guard iteration numbers when numerical linear algebra solvers are applied to solve (1.4a). These results theoretically guarantee the convergence and practically enures the efficiency for the inexact implementation of the ADMM with an easily implementable inexactness criterion. As shown by numerical results, when a standard numerical linear algebra solver is chosen for solving (1.4a), usually only a few CG steps are enough to meet the inexactness criterion to be proposed. This means it is neither necessary nor possible to solve the involved linear systems too accurately. This property significantly saves computation for solving (1.4a) and promises efficient applications of the ADMM to big-data scenarios of the model (1.1).

The resulting analysis is indeed complicated; we thus only consider the simple case where the x-subproblem in (1.4) is solved inexactly and the second one is assumed to have a closed-form solution.

[1] http://stanford.edu/~boyd/papers/admm/lasso/lasso.html

This essentially means we assume that the function $g(y)$ in the generic model (1.1) is relatively easy (e.g., the ℓ_1 penalty or some other popular penalty terms) so that the subproblem (1.4b) is easy to solve. This simplification helps us expose our main idea and analysis more clearly. As mentioned, discussing the implementation of the ADMM (1.4) to the big-data scenario of the LASSO model (1.2) is still our emphasis.

The remaining part of this paper is organized as follows. The inexactness criterion and corresponding inexact version of the ADMM are presented in Section 2. Then, we prove the convergence of the inexact version of ADMM in Section 3 and establish its worst-case convergence rate in Section 4. In Section 5, we discuss how to specify the safe-guard iteration numbers for the internally nested iterations when some standard numerical linear algebra solvers are used to solve (1.4a). In Section 6, we test some big datasets of the LASSO model (1.2) and report the numerical results. Finally, some conclusions are drawn in Section 7.

2 Algorithm

In this section we specify an inexactness criterion for (1.4a) inexactly and present an inexact version of the ADMM for the model (1.1) with special consideration of the big-data scenario where p and n are both assumed to be huge.

Let us first check the x-subproblem (1.4a). Obviously, it can be rewritten as

$$\begin{aligned} x^{k+1} &= \arg\min_{x\in\mathbb{R}^n} \mathcal{L}_\beta\left(x, y^k, \lambda^k\right) \\ &= \arg\min_{x\in\mathbb{R}^n} \left\{ \frac{1}{2}\left\|Qx-q\right\|_2^2 + \frac{\beta}{2}\left\| Ax + By^k - b - \frac{\lambda^k}{\beta}\right\|_2^2 \right\}, \end{aligned} \tag{2.1}$$

and thus x^{k+1} is given by the system of linear equations:

$$Hx^{k+1} = h^k \tag{2.2}$$

with

$$H := \left(Q^TQ + \beta A^TA\right) \quad \text{and} \quad h^k := Q^Tq - \beta A^T\left(By^k - b - \frac{\lambda^k}{\beta}\right). \tag{2.3}$$

More explicitly, we have

$$x^{k+1} = H^{-1}h^k = \left(Q^TQ + \beta A^TA\right)^{-1}\left(Q^Tq - \beta A^T\left(By^k - b - \frac{\lambda^k}{\beta}\right)\right). \tag{2.4}$$

Recall that our interest is the big-data scenario where n is huge. It is thus not preferable to directly solve the linear system (2.2) with the matrix H in dimension of $n \times n$.

Alternatively, as mentioned in [5, Section 4.2.4], we can calculate x^{k+1} via the following process:

$$\begin{cases} \hat{H}\eta^{k+1} = Q\left(\beta A^TA\right)^{-1}h^k, & \text{(2.5a)} \\ x^{k+1} = \left(\beta A^TA\right)^{-1}\left(h^k - Q^T\eta^{k+1}\right), & \text{(2.5b)} \end{cases}$$

with

$$\hat{H} := \left(Q\left(\beta A^TA\right)^{-1}Q^T + I\right). \tag{2.6}$$

To see the reason, we have

$$\begin{aligned} x^{k+1} &= \left(\beta A^T A\right)^{-1}\left(h^k - Q^T\hat{H}^{-1}Q\left(\beta A^T A\right)^{-1}h^k\right) \\ &= \left[\left(\beta A^T A\right)^{-1} - \left(\beta A^T A\right)^{-1}Q^T(Q\left(\beta A^T A\right)^{-1}Q^T + I)^{-1}Q\left(\beta A^T A\right)^{-1}\right]h^k \\ &= \left(Q^TQ + \beta A^T A\right)^{-1}h^k = H^{-1}h^k, \end{aligned} \tag{2.7}$$

where the last equality comes from the Woodbury matrix identity, see [42]. For the case with $p \ll n$, the dimension of $\hat{H} \in \mathbb{R}^{p\times p}$ in (2.6) is much smaller than $H \in \mathbb{R}^{n\times n}$ in (2.2). Hence, we prefer the procedure (2.5) than (2.4) for solving the linear system (2.2) when $p \ll n$.

To perform (2.5), we need to compute the inverse of the matrix $A^TA \in \mathbb{R}^{n\times n}$. For many applications, the coefficient matrix A is generally easier compared with the matrix Q in the objective function of the model (1.1); thus it is easier to compute $(A^TA)^{-1}$ than $(Q^TQ)^{-1}$ despite of the same dimensionality. The LASSO model (1.2) with A the identity matrix is such an application. More applications can be found in many other areas. For example, for some ℓ_1-regularized variational image restoration problems in [21], in form of (1.1), A is a circulant matrix and thus the fast Fourier transform (FFT) is applicable for efficiently solving the system of linear equations: $\left(A^TA\right)x = 0$. For such applications, computing x^{k+1} via (2.5) is much more efficient than (2.4).

For big-data scenarios of the model (1.1), even p could be still huge and hence it is not preferable to solve (2.5a) exactly by a direct method or inexactly up to a very high accuracy by an iterative method. Thus, we need to further consider how to solve the linea system (2.5a) inexactly. For this purpose, we need to specify an inexactness criterion. Obviously, the residual of the linear system (2.5a) is

$$e_k(\eta) := Q\left(\beta A^T A\right)^{-1}h^k - \hat{H}\eta. \tag{2.8}$$

At the $(k+1)$-th iteration, we suggest finding an approximate solution of the linear system (2.5a), η^{k+1}, such that

$$\left\|e_k\left(\eta^{k+1}\right)\right\|_2 \le \sigma\left\|e_k\left(\eta^k\right)\right\|_2, \tag{2.9}$$

where σ is an arbitrary constant satisfying

$$0 < \sigma < \frac{\sqrt{2\beta}}{\sqrt{2\beta} + \left\|Q\left(A^TA\right)^{-1}A^T\right\|_2} \in (0,1). \tag{2.10}$$

Note that the parameter σ plays the role of controlling the accuracy of solving the linear system (2.5a) inexactly. Obviously, larger values of σ imply looser criteria and thus less computation for solving the linear system (2.5a) inexactly; indeed in our numerical experiments, we choose values very close to the upper bound given in (2.10).

Standard numerical linear algebra solvers such as the Jacobi, Gauss-Seidel, Successive Over-Relaxation (SOR), CG and PCG methods can all be used to achieve the inexactness criterion (2.9) for the internal iterations (*i.e.*, Step 3 in the algorithm below). Accordingly, an inexactness version of the ADMM (1.4) for big-data scenario of the model (1.1) with internally nested iterations for solving the x-subproblem at each iteration in sense of the inexactness criterion (2.9) can be presented as below.

Algorithm 1 Inexact ADMM (**InADMM**) for Big-data Scenarios of (1.1)

Require: $\left(\eta^0, y^0, \lambda^0\right) \in \mathbb{R}^p \times \mathbb{R}^m \times \mathbb{R}^\ell$, $\beta > 0$;

1: Let $\hat{H}$, h^k and $e_k(\eta)$ be defined in (2.6), (2.3) and (2.8); choose σ satisfying (2.10).
2: **while** not converged **do**
3: Find η^{k+1} such that (2.9) is satisfied and compute $x^{k+1} = \left(\beta A^T A\right)^{-1}\left(h^k - Q^T \eta^{k+1}\right)$;
4: $y^{k+1} := \arg\min\limits_{y \in \mathbb{R}^m} \left\{ g(y) - \left(\lambda^k\right)^T \left(Ax^{k+1} + By - b\right) + \frac{\beta}{2}\left\|Ax^{k+1} + By - b\right\|_2^2 \right\}$;
5: $\lambda^{k+1} = \lambda^k - \beta\left(Ax^{k+1} + By^{k+1} - b\right)$.
6: **end while**

3 Convergence analysis

In this section, we prove the convergence of the proposed InADMM. As mentioned, though the convergence of the original ADMM in the ideal exact form of (1.4) has been well studied in the literature, these existing results cannot be directly extended to the proposed InADMM because of the specific inexact steps for calculating x^{k+1}. We hence present the complete analysis for the convergence of InADMM. We start from some known preliminaries.

3.1 Preliminaries

Let $\Omega := \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^\ell$. To present our analysis in more compact notation, we denote the vectors $w \in \Omega$ and $u \in \mathbb{R}^m \times \mathbb{R}^\ell$, the matrix $M \in \mathbb{R}^{(m+\ell)\times(m+\ell)}$ and the function $F(w)$ as following:

$$w = \begin{pmatrix} x \\ y \\ \lambda \end{pmatrix}, \quad u = \begin{pmatrix} y \\ \lambda \end{pmatrix}, \quad \mathcal{M} = \begin{pmatrix} \beta B^T B & 0 \\ 0 & \frac{1}{\beta} I_{\ell\times\ell} \end{pmatrix}, \quad F(w) = \begin{pmatrix} Q^T\left(Qx - q\right) - A^T\lambda \\ -B^T\lambda \\ Ax + By - b \end{pmatrix}. \tag{3.1}$$

Note that the matrix $\mathcal{M}$ is not necessarily positive definite because the matrix B is not assumed to be full column rank in (1.1). As follows, we slightly abuse the notation $\|u\|_{\mathcal{M}}$ to denote the number $\sqrt{u^T\mathcal{M}u}$.

As initiated in [24], the problem (1.1) can be stated as the variational inequality of finding $w^\star = (x^\star, y^\star, \lambda^\star) \in \Omega$ such that

$$\mathrm{VI}(\Omega, F): \quad g(y) - g(y^\star) + \left(w - w^\star\right)^T F(w^\star) \geq 0, \ \forall w \in \Omega. \tag{3.2}$$

The solution set of the variational inequality (3.2) is denoted by $\Omega^\star$. As analyzed in [13, 24], the solution set $\Omega^\star$ has the characterization shown in the following theorem. We skip the proof, which can be found in [24] as well.

Theorem 3.1. *Let $\Omega^\star$ be the solution set of the variational inequality (3.2). Then, we have*

$$\Omega^\star = \bigcap_{w\in\Omega} \left\{ \hat{w} \in \Omega : g(y) - g(\hat{y}) + \left(w - \hat{w}\right)^T F\left(w\right) \geq 0 \right\}. \tag{3.3}$$

3.2 Optimality

To derive the convergence of the proposed InADMM, it is necessary to discern the difference of its iterate from a solution point, or the optimality of each iterate. More specifically, we need to quantity how accurately the x-subproblem in the ideal (exact) form of the ADMM (1.4) is approached by the inexact step (*i.e.*, Step 3) of the proposed InADMM.

For this purpose, with the x^{k+1} generated by the InADMM, we have

$$\begin{aligned}\nabla_x \mathcal{L}_\beta \left(x^{k+1}, y^k, \lambda^k\right) &= \left(Q^T Q + \beta A^T A\right) x^{k+1} - h^k \\ &= \left(Q^T Q + \beta A^T A\right) \left(\beta A^T A\right)^{-1} \left(h^k - Q^T \eta^{k+1}\right) - h^k \\ &= Q^T \left[Q \left(\beta A^T A\right)^{-1} h^k - \left(Q \left(\beta A^T A\right)^{-1} Q^T + I\right) \eta^{k+1}\right] \\ &= Q^T e_k \left(\eta^{k+1}\right).\end{aligned}$$

For the exact version of ADMM (1.4) where x is required to be solved exactly, it obviously means $\nabla_x \mathcal{L}_\beta \left(x, y^k, \lambda^k\right) = 0$. Therefore, we can quantitatively regard $\|Q^T e_k \left(\eta^{k+1}\right)\|_2$ as the difference of the inexact solution of the x-subproblem generated by the InADMM from the exact solution generated by the exact version of ADMM (1.4), in sense of the residual of the partial gradient of the augmented Lagrangian function. Recall that the y- and λ-subproblems are assumed to be solved exactly in the InADMM. Hence, for the iterate $w^{k+1} = (x^{k+1}, y^{k+1}, \lambda^{k+1})$ generate by the InADMM, its optimality can be expressed as the following:

$$\begin{cases} \nabla_x \mathcal{L}_\beta \left(x^{k+1}, y^k, \lambda^k\right) = Q^T e_k(\eta^{k+1}), \\ g(y) - g\left(y^{k+1}\right) + \left(y - y^{k+1}\right)^T \left(-B^T \lambda^k + \beta B^T \left(Ax^{k+1} + By^{k+1} - b\right)\right) \geq 0, \ \forall y \in \mathbb{R}^m, \\ \lambda^{k+1} = \lambda^k - \beta \left(Ax^{k+1} + By^{k+1} - b\right). \end{cases} \tag{3.4}$$

We reiterate that if $\|e_k(\eta^{k+1})\|_2 = 0$, then the InADMM reduces to the exact version of ADMM (1.4) with known convergence. But we focus on the big-data scenarios where it is not possible to pursue $\|e_k(\eta^{k+1})\|_2 = 0$ and only an approximate solution subject to the inexactness criterion (2.9) is realized by internally nested iterations.

To prove the convergence of the sequence generated by the InADMM whose optimality is given in (3.4), it is also crucial to analyze how the residual $\|e_k(\eta^{k+1})\|_2$ evolves according to the iterations. Recall that the y- and λ-subproblems are assumed to be solved exactly in the InADMM. We thus know that

$$B^T \lambda^{k-1} \in \partial g(y^{k-1}), \ \ B^T \lambda^k \in \partial g(y^k),$$

and

$$\left(y^{k-1} - y^k\right)^T B^T \left(\lambda^{k-1} - \lambda^k\right) \geq 0. \tag{3.5}$$

Hence, it is easily derived that

$$\left\|\lambda^{k-1} - \lambda^k - \beta B \left(y^{k-1} - y^k\right)\right\|_2^2 \leq \left\|\lambda^{k-1} - \lambda^k\right\|_2^2 + \beta^2 \left\|B \left(y^{k-1} - y^k\right)\right\|_2^2 = \beta \left\|u^{k-1} - u^k\right\|_{\mathcal{M}}^2, \tag{3.6}$$

where $\mathcal{M}$ is defined in (3.1). As a result, we have

$$\begin{aligned}
\left\|e_k(\eta^{k+1})\right\|_2 &\leq \sigma\left\|e_k(\eta^k)\right\|_2 = \sigma\left\|Q\left(\beta A^TA\right)^{-1}h^k - \hat{H}\eta^k\right\|_2 \\
&\leq \sigma\left\|Q\left(\beta A^TA\right)^{-1}h^{k-1} - \hat{H}\eta^k\right\|_2 + \sigma\left\|Q\left(\beta A^TA\right)^{-1}\left(h^k - h^{k-1}\right)\right\|_2 \\
&\overset{h^{k-1},h^k}{\leq} \sigma\left\|e_{k-1}(\eta^k)\right\|_2 + \sigma\left\|Q\left(\beta A^TA\right)^{-1}A^T\right\|_2 \cdot \left\|\beta B\left(y^{k-1}-y^k\right) - \left(\lambda^{k-1}-\lambda^k\right)\right\|_2 \\
&\overset{(3.6)}{\leq} \sigma\left\|e_{k-1}(\eta^k)\right\|_2 + \frac{\sigma}{\sqrt{\beta}}\left\|Q(A^TA)^{-1}A^T\right\|_2 \cdot \left\|u^{k-1}-u^k\right\|_{\mathcal{M}}. \qquad (3.7)
\end{aligned}$$

Therefore, for two consecutive iterates generated by the InADMM, it follows from (3.7) that their residuals arising from solving the x-subproblems inexactly are related precisely by

$$\left\|e_k(\eta^{k+1})\right\|_2 \leq \sigma\left\|e_{k-1}(\eta^k)\right\|_2 + \sigma\gamma\left\|u^{k-1}-u^k\right\|_{\mathcal{M}} \qquad (3.8)$$

with

$$\gamma = \frac{\left\|Q\left(A^TA\right)^{-1}A^T\right\|_2}{\sqrt{\beta}}. \qquad (3.9)$$

This relationship will be often used in the coming analysis.

Moreover, recall that the parameter σ controlling the accuracy in (2.9) is restricted by the condition (2.10). Hence, it follows from the definition of γ in (3.9) that

$$0 < \frac{\gamma^2\sigma^2}{2(1-\sigma)^2} = \left(\frac{\sigma}{2(1-\sigma)}\right)\left(\frac{\gamma^2\sigma}{1-\sigma}\right) < 1$$

and obviously there exists $\mu > 0$ such that

$$0 < \frac{\mu}{2}\frac{\sigma}{1-\sigma} < 1 \quad \text{and} \quad 0 < \frac{\gamma^2}{\mu}\frac{\sigma}{1-\sigma} < 1. \qquad (3.10)$$

3.3 Convergence

Now we prove the convergence of the sequence generated by the InADMM. To simplify the notation, let us introduce an auxiliary variable $\bar{w}^k$ as

$$\bar{w}^k = \begin{pmatrix}\bar{x}^k\\ \bar{y}^k\\ \bar{\lambda}^k\end{pmatrix} = \begin{pmatrix}x^{k+1}\\ y^{k+1}\\ \lambda^k - \beta(Ax^{k+1}+By^k-b)\end{pmatrix}. \qquad (3.11)$$

This notation is only for the simplification of notation in our analysis; it is not required to be computed for implementing the InADMM.

Recall the variational inequality reformulation (3.2) of the model (1.1). First of all, we analyze how different the point $\bar{w}^k$ defined in (3.11) is from a solution point of (3.2); and how to quantify this difference by iterates generated by the InADMM.

Proposition 3.2. *Let $\{w^k\}$ be the sequence generated by the InADMM; $\bar{w}^k$ be defined in (3.11) and $\mathcal{M}$ in (3.1). Then, for all $w \in \Omega$, it holds that*

$$\begin{aligned}
g\left(\bar{y}^k\right) - g(y) + \left(\bar{w}^k - w\right)^T F\left(\bar{w}^k\right) \;\leq\; & \frac{1}{2}\left(\left\|u^k-u\right\|_{\mathcal{M}}^2 - \left\|u^{k+1}-u\right\|_{\mathcal{M}}^2 - \left\|u^k-u^{k+1}\right\|_{\mathcal{M}}^2\right) \\
& + \left(x^{k+1}-x\right)^T\nabla_x\mathcal{L}_\beta\left(x^{k+1},y^k,\lambda^k\right). \qquad (3.12)
\end{aligned}$$

Proof. First we rewrite $\nabla_x \mathcal{L}_\beta \left(x^{k+1}, y^k, \lambda^k\right)$ as

$$\begin{aligned}\nabla_x \mathcal{L}_\beta \left(x^{k+1}, y^k, \lambda^k\right) &= Q^T \left(Qx^{k+1} - q\right) - A^T \left(\lambda^k - \beta \left(Ax^{k+1} + By^k - b\right)\right) \\ &= Q^T \left(Qx^{k+1} - q\right) - A^T \bar{\lambda}^k.\end{aligned}$$

Then, combining it with the optimality condition with respect to y, for all $y \in \mathbb{R}^m$, we have

$$g(y) - g\left(y^{k+1}\right) + \left(y - y^{k+1}\right)\left[-B^T\lambda^k + \beta B^T\left(Ax^{k+1} + By^{k+1} - b\right)\right] \geq 0, \tag{3.13}$$

with which we obtain, for all $w \in \Omega$, that

$$\begin{aligned}& g(y) - g\left(\bar{y}^k\right) + \left(w - \bar{w}^k\right)^T F\left(\bar{w}^k\right) \\ = \;& \left(x - x^{k+1}\right)^T \left(Q^T\left(Qx^{k+1} - q\right) - A^T\bar{\lambda}^k\right) + g(y) - g\left(y^{k+1}\right) + \left(y - y^{k+1}\right)^T\left(-B^T\bar{\lambda}^k\right) + \\ & \left(\lambda - \bar{\lambda}^k\right)\left(Ax^{k+1} + By^{k+1} - b\right) \\ = \;& \left(x - x^{k+1}\right)^T \nabla_x \mathcal{L}_\beta\left(x^{k+1}, y^k, \lambda^k\right) + \left(y - y^{k+1}\right)\left[-B^T\lambda^k + \beta B^T\left(Ax^{k+1} + By^{k+1} - b\right)\right] \\ & + g(y) - g\left(y^{k+1}\right) + \beta\left(y - y^{k+1}\right)B^TB\left(y^k - y^{k+1}\right) + \frac{1}{\beta}\left(\lambda - \bar{\lambda}^k\right)^T\left(\lambda^k - \lambda^{k+1}\right) \\ \overset{(3.13)}{\geq} \;& \left(x - x^{k+1}\right)^T \nabla_x \mathcal{L}_\beta\left(x^{k+1}, y^k, \lambda^k\right) + \beta\left(y - y^{k+1}\right)B^TB\left(y^k - y^{k+1}\right) \\ & + \frac{1}{\beta}\left(\lambda - \lambda^{k+1}\right)^T\left(\lambda^k - \lambda^{k+1}\right) + \frac{1}{\beta}\left(\lambda^{k+1} - \bar{\lambda}^k\right)^T\left(\lambda^k - \lambda^{k+1}\right).\end{aligned} \tag{3.14}$$

Moreover, notice the elementary equation

$$(a - c)^T(b - c) = \frac{1}{2}\left(\|a - c\|_2^2 - \|a - b\|_2^2 + \|b - c\|_2^2\right). \tag{3.15}$$

Thus, for all $w \in \Omega$, we have

$$\begin{aligned}& g(y) - g\left(\bar{y}^k\right) + \left(w - \bar{w}^k\right)^T F\left(\bar{w}^k\right) \\ \overset{(3.15)}{\geq} \;& \left(x - x^{k+1}\right)^T \nabla_x \mathcal{L}_\beta\left(x^{k+1}, y^k, \lambda^k\right) + \frac{\beta}{2}\left(\left\|B\left(y - y^{k+1}\right)\right\|_2^2 - \left\|B\left(y - y^k\right)\right\|_2^2 + \left\|B\left(y^k - y^{k+1}\right)\right\|_2^2\right) \\ & + \frac{1}{2\beta}\left(\left\|\lambda - \lambda^{k+1}\right\|_2^2 - \left\|\lambda - \lambda^k\right\|_2^2 + \left\|\lambda^k - \lambda^{k+1}\right\|_2^2\right) + \left(y^k - y^{k+1}\right)^T B^T\left(\lambda^k - \lambda^{k+1}\right) \\ \overset{(3.5)}{\geq} \;& \left(x - x^{k+1}\right)^T \nabla_x \mathcal{L}_\beta\left(x^{k+1}, y^k, \lambda^k\right) + \frac{\beta}{2}\left(\left\|B\left(y - y^{k+1}\right)\right\|_2^2 - \left\|B\left(y - y^k\right)\right\|_2^2\right) \\ & + \frac{1}{2\beta}\left(\left\|\lambda - \lambda^{k+1}\right\|_2^2 - \left\|\lambda - \lambda^k\right\|_2^2\right) + \frac{1}{2}\left\|u^k - u^{k+1}\right\|_{\mathcal{M}}^2.\end{aligned} \tag{3.16}$$

Using the notation of $\mathcal{M}$ in (3.1), (3.16) can be rewritten as (3.12) and the proof is complete. □

The difference between the inequality (3.12) and the variational inequality reformulation (3.2) reflects the difference of the point $\bar{w}^k$ from a solution point $w^\star$. For the right-hand side of (3.12), the first three terms are quadratic and they are easy to be manipulated over different indicators by algebraic operations, but it is not that explicit how the last crossing term can be controlled towards the eventual goal of proving the convergence of the sequence $\{w^k\}$. We thus look into this term particularly and show that the sum of these crossing terms over K iterations can be bounded by some quadratic terms as well. This result is summarized in the following proposition.

Proposition 3.3. *Let $\{w^k\}$ be the sequence generated by InADMM. For all $x \in \mathbb{R}^n$, $K > 1$ and μ satisfying (3.10), it holds that*

$$\begin{aligned}
&\sum_{k=1}^{K} \left(x^{k+1} - x\right)^T \nabla_x \mathcal{L}_\beta \left(x^{k+1}, y^k, \lambda^k\right) \\
&\leq \frac{\sigma}{1-\sigma} \left\{ \frac{\mu}{2} \sum_{k=1}^{K} \left\| Q\left(x^{k+1} - x\right) \right\|_2^2 + \frac{1}{2\mu} \left[\sum_{k=1}^{K-1} \gamma^2 \left\| u^k - u^{k+1} \right\|_{\mathcal{M}}^2 + \left(\left\| e_0\left(\eta^1\right) \right\|_2 + \gamma \left\| u^0 - u^1 \right\|_{\mathcal{M}} \right)^2 \right] \right\}.
\end{aligned} \tag{3.17}$$

Proof. Recall the result (3.8). By mathematical induction, for all $k \geq 1$, we have

$$\left\| e_k(\eta^{k+1}) \right\|_2 \leq \sum_{i=0}^{k-1} \sigma^{k-i} \gamma \left\| u^i - u^{i+1} \right\|_{\mathcal{M}} + \sigma^k \left\| e_0(\eta^1) \right\|_2. \tag{3.18}$$

Then, combining it the optimality condition with respect to the x-subproblem at each iteration, we have that, for all $x \in \mathbb{R}^n$ and $K > 1$, the following inequality holds:

$$\begin{aligned}
&\sum_{k=1}^{K} \left(x^{k+1} - x\right)^T \nabla_x \mathcal{L}_\beta \left(x^{k+1}, y^k, \lambda^k\right) = \sum_{k=1}^{K} \left(x^{k+1} - x\right)^T Q^T e_k(\eta^{k+1}) \\
\leq\ & \sum_{k=1}^{K} \left\| Q\left(x^{k+1} - x\right) \right\|_2 \cdot \left\| e_k(\eta^{k+1}) \right\|_2 \\
\overset{(3.18)}{\leq}\ & \sum_{k=1}^{K} \sum_{i=0}^{k-1} \sigma^{k-i} \gamma \left\| Q\left(x^{k+1} - x\right) \right\|_2 \cdot \left\| u^i - u^{i+1} \right\|_{\mathcal{M}} + \sum_{k=1}^{K} \sigma^k \left\| Q\left(x^{k+1} - x\right) \right\|_2 \cdot \left\| e_0(\eta^1) \right\|_2 \\
=\ & \sum_{k=2}^{K} \sum_{i=1}^{k-1} \sigma^{k-i} \gamma \left\| Q\left(x^{k+1} - x\right) \right\|_2 \cdot \left\| u^i - u^{i+1} \right\|_{\mathcal{M}} + \sum_{k=1}^{K} \sigma^k \left\| Q\left(x^{k+1} - x\right) \right\|_2 \left(\left\| e_0(\eta^1) \right\|_2 + \gamma \left\| u^0 - u^1 \right\|_{\mathcal{M}} \right) \\
\leq\ & \frac{\mu}{2} \sum_{k=2}^{K} \sum_{i=1}^{k-1} \sigma^{k-i} \left\| Q\left(x^{k+1} - x\right) \right\|_2^2 + \frac{1}{2\mu} \sum_{k=2}^{K} \sum_{i=1}^{k-1} \sigma^{k-i} \gamma^2 \left\| u^i - u^{i+1} \right\|_{\mathcal{M}}^2 \\
&\qquad + \frac{\mu}{2} \sum_{k=1}^{K} \sigma^k \left\| Q\left(x^{k+1} - x\right) \right\|_2^2 + \frac{1}{2\mu} \sum_{k=1}^{K} \sigma^k \left(\left\| e_0(\eta^1) \right\|_2 + \gamma \left\| u^0 - u^1 \right\|_{\mathcal{M}} \right)^2 \\
=\ & \frac{\mu}{2} \sum_{k=1}^{K} \sum_{i=0}^{k-1} \sigma^{k-i} \left\| Q\left(x^{k+1} - x\right) \right\|_2^2 + \frac{1}{2\mu} \sum_{k=2}^{K} \sum_{i=1}^{k-1} \sigma^{k-i} \gamma^2 \left\| u^i - u^{i+1} \right\|_{\mathcal{M}}^2 \\
&\qquad + \frac{1}{2\mu} \sum_{k=1}^{K} \sigma^k \left[\left\| e_0(\eta^1) \right\|_2 + \gamma \left\| u^0 - u^1 \right\|_{\mathcal{M}} \right]^2.
\end{aligned}$$

Furthermore, for all $x \in \mathbb{R}^n$ and $K > 1$, we have

$$\sum_{k=1}^{K} \sum_{i=0}^{k-1} \sigma^{k-i} \left\| Q\left(x^{k+1} - x\right) \right\|_2^2 \overset{\sigma \in (0,1)}{=} \sum_{k=1}^{K} \frac{\sigma - \sigma^{k+1}}{1-\sigma} \left\| Q\left(x^{k+1} - x\right) \right\|_2^2, \tag{3.19}$$

and

$$\begin{aligned}
\sum_{k=2}^{K} \sum_{i=1}^{k-1} \sigma^{k-i} \gamma^2 \left\| u^i - u^{i+1} \right\|_{\mathcal{M}}^2 &= \sum_{i=1}^{K-1} \sum_{k=i+1}^{K} \sigma^{k-i} \gamma^2 \left\| u^i - u^{i+1} \right\|_{\mathcal{M}}^2 \\
&\overset{\sigma \in (0,1)}{=} \sum_{i=1}^{K-1} \frac{\sigma - \sigma^{K-i+1}}{1-\sigma} \gamma^2 \left\| u^i - u^{i+1} \right\|_{\mathcal{M}}^2.
\end{aligned} \tag{3.20}$$

Combining the above equalities and inequalities, we obtain

$$\begin{aligned}
&\sum_{k=1}^{K}\left(x^{k+1}-x\right)^{T}\nabla_{x}\mathcal{L}_{\beta}\left(x^{k+1},y^{k},\lambda^{k}\right)\\
\overset{(3.19)(3.20)}{=}\quad&\frac{\mu}{2}\sum_{k=1}^{K}\frac{\sigma-\sigma^{k+1}}{1-\sigma}\left\|Q\left(x^{k+1}-x\right)\right\|_{2}^{2}+\frac{1}{2\mu}\sum_{i=1}^{K-1}\frac{\sigma-\sigma^{K-i+1}}{1-\sigma}\gamma^{2}\left\|u^{i}-u^{i+1}\right\|_{\mathcal{M}}^{2}\\
&+\frac{1}{2\mu}\frac{\sigma-\sigma^{K+1}}{1-\sigma}\left(\left\|e_{0}(\eta^{1})\right\|_{2}+\gamma\left\|u^{0}-u^{1}\right\|_{\mathcal{M}}\right)^{2}\\
\leq\quad&\frac{\mu}{2}\sum_{k=1}^{K}\frac{\sigma}{1-\sigma}\left\|Q\left(x^{k+1}-x\right)\right\|_{2}^{2}+\frac{1}{2\mu}\sum_{i=1}^{K-1}\frac{\sigma}{1-\sigma}\gamma^{2}\left\|u^{i}-u^{i+1}\right\|_{\mathcal{M}}^{2}\\
&+\frac{1}{2\mu}\frac{\sigma}{1-\sigma}\left(\left\|e_{0}(\eta^{1})\right\|_{2}+\gamma\left\|u^{0}-u^{1}\right\|_{\mathcal{M}}\right)^{2},
\end{aligned}$$

which implies the conclusion (3.17). The proof is complete. □

The convergence of the proposed InADMM is established in the following theorem.

Theorem 3.4. *Let $\{w^k\}$ be the sequence generated by the InADMM. Then, we have the following assertions:*

(1) $\left\|e_k(\eta^{k+1})\right\|_2 \overset{k\to\infty}{\longrightarrow} 0, \quad \left\|B\left(y^k-y^{k+1}\right)\right\|_2 \overset{k\to\infty}{\longrightarrow} 0$;

(2) $\left\|Ax^{k+1}+By^{k+1}-b\right\|_2 \overset{k\to\infty}{\longrightarrow} 0$; $f\left(x^k\right)+g\left(y^k\right) \overset{k\to\infty}{\longrightarrow} f\left(x^\star\right)+g\left(y^\star\right)$ *for any given* $w^\star\in\Omega^\star$.

Proof. First, recall the definition of $F(w)$ in (3.1). We have that

$$\left(w-\bar{w}^k\right)^{T}\left(F\left(w\right)-F\left(\bar{w}^k\right)\right)=\left(x-x^{k+1}\right)^{T}Q^{T}\left(Qx-Qx^{k+1}\right)=\left\|Q\left(x-x^{k+1}\right)\right\|_{2}^{2}. \tag{3.21}$$

Then, using the results (3.12) and (3.17) established in Propositions 3.2 and 3.3, respectively, we obtain

$$\begin{aligned}
&\sum_{k=1}^{K}\left\{g\left(\bar{y}^k\right)-g\left(y\right)+\left(\bar{w}^k-w\right)^{T}F\left(w\right)\right\}\\
=\quad&\sum_{k=1}^{K}\left\{g\left(\bar{y}^k\right)-g\left(y\right)+\left(\bar{w}^k-w\right)^{T}F\left(\bar{w}^k\right)+\left(\bar{w}^k-w\right)^{T}\left[F\left(w\right)-F\left(\bar{w}^k\right)\right]\right\}\\
\overset{(3.12)}{\leq}\quad&\frac{1}{2}\left(\left\|u^{1}-u\right\|_{\mathcal{M}}^{2}-\left\|u^{K+1}-u\right\|_{\mathcal{M}}^{2}\right)+\sum_{k=1}^{K}\left\{\left(x^{k+1}-x\right)^{T}\nabla_{x}\mathcal{L}_{\beta}\left(x^{k+1},y^{k},\lambda^{k}\right)\right.\\
&\left.-\left(w-\bar{w}^k\right)^{T}\left[F\left(w\right)-F\left(\bar{w}^k\right)\right]\right\}-\sum_{k=1}^{K}\frac{1}{2}\left\|u^{k}-u^{k+1}\right\|_{\mathcal{M}}^{2}\\
\overset{(3.17)(3.21)}{\leq}\quad&\frac{1}{2}\left(\left\|u^{1}-u\right\|_{\mathcal{M}}^{2}-\left\|u^{K+1}-u\right\|_{\mathcal{M}}^{2}\right)+\sum_{k=1}^{K}\left(\frac{\mu}{2}\frac{\sigma}{1-\sigma}-1\right)\left\|Q\left(x^{k+1}-x\right)\right\|_{2}^{2}\\
&+\sum_{k=1}^{K-1}\frac{1}{2}\left(\frac{\sigma}{1-\sigma}\frac{\gamma^{2}}{\mu}-1\right)\left\|u^{k}-u^{k+1}\right\|_{\mathcal{M}}^{2}-\frac{1}{2}\left\|u^{K}-u^{K+1}\right\|_{\mathcal{M}}^{2}\\
&+\frac{1}{2\mu}\frac{\sigma}{1-\sigma}\left(\left\|e_{0}(\eta^{1})\right\|_{2}+\gamma\left\|u^{0}-u^{1}\right\|_{\mathcal{M}}\right)^{2}.
\end{aligned}\tag{3.22}$$

For any given $w^\star\in\Omega^\star$, we have

$$g\left(\bar{y}^k\right)-g\left(y^\star\right)+\left(\bar{w}^k-w^\star\right)^{T}F\left(w^\star\right)\geq 0,\ \forall k.$$

Setting $w = w^\star$ in (3.22), together with the above property, for any $K > 1$, we have

$$\begin{aligned}
&\sum_{k=1}^{K}\left(1-\frac{\mu}{2}\frac{\sigma}{1-\sigma}\right)\left\|Q\left(x^{k+1}-x^\star\right)\right\|_2^2+\sum_{k=1}^{K-1}\left(\frac{1}{2}-\frac{\gamma^2}{2\mu}\frac{\sigma}{1-\sigma}\right)\left\|u^k-u^{k+1}\right\|_{\mathcal{M}}^2\\
\leq\ & \frac{1}{2}\left\|u^1-u^\star\right\|_{\mathcal{M}}^2+\frac{1}{2\mu}\frac{\sigma}{1-\sigma}\left(\left\|e_0(\eta^1)\right\|_2+\gamma\left\|u^0-u^1\right\|_{\mathcal{M}}\right)^2-\frac{1}{2}\left\|u^{K+1}-u^\star\right\|_{\mathcal{M}}^2-\frac{1}{2}\left\|u^K-u^{K+1}\right\|_{\mathcal{M}}^2.
\end{aligned} \tag{3.23}$$

Using (3.10), we conclude that

$$\left\|Q\left(x^{k+1}-x^\star\right)\right\|_2 \overset{k\to\infty}{\longrightarrow} 0, \quad \left\|u^k-u^{k+1}\right\|_{\mathcal{M}} \overset{k\to\infty}{\longrightarrow} 0 \quad \text{and} \quad \left\|u^{K+1}-u^\star\right\|_{\mathcal{M}}<\infty. \tag{3.24}$$

Furthermore, for any $\varepsilon > 0$, there exists k_0 such that for all $k \geq k_0$, we have

$$\left\|u^k-u^{k+1}\right\|_{\mathcal{M}} \leq \varepsilon \quad \text{and} \quad \sigma^k \leq \varepsilon.$$

For all $k > 2k_0$, it follows from (3.18) that

$$\begin{aligned}
\left\|e_k(\eta^{k+1})\right\|_2 &\leq \sum_{i=0}^{k-1}\sigma^{k-i}\gamma\left\|u^i-u^{i+1}\right\|_{\mathcal{M}}+\sigma^k\left\|e_0(\eta^1)\right\|_2\\
&= \sum_{i=0}^{k_0-1}\sigma^{k-i}\gamma\left\|u^i-u^{i+1}\right\|_{\mathcal{M}}+\sum_{k_0}^{k-1}\sigma^{k-i}\gamma\left\|u^i-u^{i+1}\right\|_{\mathcal{M}}+\sigma^k\left\|e_0(\eta^1)\right\|_2\\
&\leq \left(\max_{0\leq i\leq k_0-1}\left\{\left\|u^i-u^{i+1}\right\|_{\mathcal{M}}\right\}\gamma\sum_{i=0}^{k_0-1}\sigma^{k-k_0-i}\right)\cdot\sigma^{k_0}+\sigma^k\left\|e_0(\eta^1)\right\|_2\\
&\quad+\left(\sum_{k_0}^{k-1}\sigma^{k-i}\gamma\right)\cdot\max_{k_0\leq i\leq k-1}\left\{\left\|u^i-u^{i+1}\right\|_{\mathcal{M}}\right\}\\
&\leq \left[\left(\max_{0\leq i\leq k_0-1}\left\{\left\|u^i-u^{i+1}\right\|_{\mathcal{M}}\right\}\gamma\sum_{i=0}^{k_0-1}\sigma^{k-k_0-i}\right)+\left(\sum_{k_0}^{k-1}\sigma^{k-i}\gamma\right)+\left\|e_0(\eta^1)\right\|_2\right]\cdot\varepsilon,
\end{aligned}$$

which implies that

$$\left\|e_k(\eta^{k+1})\right\|_2 \overset{k\to\infty}{\longrightarrow} 0. \tag{3.25}$$

Moreover, note that $\left\|B\left(y^k-y^{k+1}\right)\right\|_2 \overset{k\to\infty}{\longrightarrow} 0$ can be obtained by the fact $\left\|u^k-u^{k+1}\right\|_{\mathcal{M}} \overset{k\to\infty}{\longrightarrow} 0$. The first assertion is proved.

Now we prove the second assertion. For the first part: $\left\|Ax^{k+1}+By^{k+1}-b\right\|_2 \overset{k\to\infty}{\longrightarrow} 0$, it follows immediately from the facts $\left\|Ax^{k+1}+By^{k+1}-b\right\|_2 = \frac{1}{\beta}\left\|\lambda^k-\lambda^{k+1}\right\|_2$ and $\left\|u^k-u^{k+1}\right\|_{\mathcal{M}} \overset{k\to\infty}{\longrightarrow} 0$. Note that the optimality conditions of the y-subproblem at the $(k+1)$-th iteration and a solution point $y^\star$ can be respectively written as

$$\begin{cases} g(y)-g\left(y^{k+1}\right)+\left(y-y^{k+1}\right)^T\left(-B^T\lambda^{k+1}\right)\geq 0,\\ g(y)-g\left(y^\star\right)+\left(y-y^\star\right)^T\left(-B^T\lambda^\star\right)\geq 0. \end{cases}$$

Accordingly, taking $y = y^\star$ and $y = y^{k+1}$ respectively in the above inequalities, we have

$$\left(y^{k+1}-y^\star\right)^T B^T\lambda^\star \leq g\left(y^{k+1}\right)-g\left(y^\star\right) \leq \left(y^{k+1}-y^\star\right)^T B^T\lambda^{k+1}. \tag{3.26}$$

The same technique can also be applied to the x-subproblem and a solution point $x^\star$. Additionally, using the convexity of f, we have

$$\begin{aligned}\left(x^{k+1}-x^\star\right)^T A^T\lambda^\star &\leq f\left(x^{k+1}\right)-f\left(x^\star\right)\\ &\leq \left(x^{k+1}-x^\star\right)^T Q^T\left(Qx^{k+1}-q\right)\\ &= \left(x^{k+1}-x^\star\right)^T\left[A^T\left(\lambda^k-\beta\left(Ax^{k+1}+By^k-b\right)\right)+Q^T e_k\left(\eta^{k+1}\right)\right]. \end{aligned} \tag{3.27}$$

Then, summarizing (3.26) and (3.27), we obtain

$$\begin{aligned}&\frac{1}{\beta}\left(\lambda^k-\lambda^{k+1}\right)^T\lambda^\star\\ =\;& \left(x^{k+1}-x^\star\right)^T A^T\lambda^\star+\left(y^{k+1}-y^\star\right)^T B^T\lambda^\star\\ \leq\;& \left[f\left(x^{k+1}\right)+g\left(y^{k+1}\right)\right]-\left[g\left(y^\star\right)+f\left(x^\star\right)\right]\\ \leq\;& \left(y^{k+1}-y^\star\right)^T B^T\lambda^{k+1}+\left(x^{k+1}-x^\star\right)^T\left[A^T\left(\lambda^k-\beta\left(Ax^{k+1}+By^k-b\right)\right)+Q^T e_k\left(\eta^{k+1}\right)\right]\\ \leq\;& \frac{1}{\beta}\left(\lambda^k-\lambda^{k+1}\right)^T\lambda^{k+1}+\beta\left(x^{k+1}-x^\star\right)^T A^T B\left(y^{k+1}-y^k\right)+\left(x^{k+1}-x^\star\right)^T Q^T e_k\left(\eta^{k+1}\right). \end{aligned} \tag{3.28}$$

Since

$$\left\|e_k(\eta^{k+1})\right\|_2\to 0,\ \left\|Qx^{k+1}-Qx^\star\right\|_2\to 0,\ \left\|u^k-u^{k+1}\right\|_{\mathcal{M}}\to 0,\ \left\|u^{K+1}-u^\star\right\|_{\mathcal{M}}<\infty,$$

and

$$A\left(x^{k+1}-x^\star\right)=\frac{1}{\beta}\left(\lambda^k-\lambda^{k+1}\right)-B\left(y^{k+1}-y^\star\right)\quad\Rightarrow\quad\left\|A\left(x^{k+1}-x^\star\right)\right\|_2^2<\infty,$$

both the left- and right-hand sides of (3.28) converge to zero. As a result, we have

$$f\left(x^{k+1}\right)+g\left(y^{k+1}\right)\overset{k\to\infty}{\longrightarrow} f\left(x^\star\right)+g\left(y^\star\right),$$

which is the second assertion of this theorem. The proof is complete. □

3.4 More discussions

In Theorem 3.4, the convergence of the proposed InADMM is established in the sense that the constraint violation converges to zero and the objective function values converge to the optimal value. The key is that our inexactness criterion (2.9) ensures that the error arising from solving the x-subproblems inexactly vanishes as the iteration goes on, *i.e.*, $\left\|e_k(\eta^{k+1})\right\|_2\overset{k\to\infty}{\longrightarrow}0$ as proved in Theorem 3.4. Hence, the proposed InADMM inherits the convergence of the exact version of ADMM (1.4), even though its x-subproblems are solved inexactly. With stronger assumptions on the model (1.1), stronger convergence results can be derived. Such a topic has its own interests and it has been widely studied in the ADMM literature. Discussing such extensions is beyond the scope of this paper; hence we just briefly mention some such extensions in this subsection and focus on establishing the convergence of the InADMM in terms of iterates under additional assumptions.

Theorem 3.5. *Let $\left\{w^k\right\}$ be the sequence generated by the InADMM. If B is further assumed to be full column rank in (1.1), then the convergence of InADMM holds in sense of its iterates, i.e.,*

$$\left\{x^k\right\}\overset{k\to\infty}{\longrightarrow}x^\infty,\quad\left\{y^k\right\}\overset{k\to\infty}{\longrightarrow}y^\infty,\quad\left\{\lambda^k\right\}\overset{k\to\infty}{\longrightarrow}\lambda^\infty,$$

where $w^\infty\in\Omega^\star$.

Proof. Recall the proof of Theorem 3.4. If B is full column rank, then the matrix $\mathcal{M}$ defined in (3.1) is positive definite. Thus, it follows from (3.24) that $\left\{u^k\right\}_{i=1}^{\infty}$ is bounded. Hence, there exists a subsequence $\left\{u^{k_i}\right\}_{i=1}^{\infty}$ converging to $u^\infty = (y^\infty, \lambda^\infty)$. Let us define $x^\infty := \left(A^TA\right)^{-1}A^T\left(b - By^\infty\right)$. Specifying Step 5 of the InADMM for the k-th iteration, we have

$$Ax^k = \frac{1}{\beta}\left(\lambda^{k-1} - \lambda^k\right) - \left(By^k - b\right),$$

which can be equivalently written as

$$x^k = \left(A^TA\right)^{-1}\left[A^T\left(\frac{1}{\beta}\left(\lambda^{k-1} - \lambda^k\right) - \left(By^k - b\right)\right)\right], \tag{3.29}$$

because A is assumed to be full column rank. According to the second assertion in Theorem 3.4, we have $\left\|\lambda^{k+1} - \lambda^k\right\|_2 \overset{i\to\infty}{\longrightarrow} 0$, together with the convergence of $\left\{y^{k_i}\right\}$ to y^∞, we know that (3.29) implies

$$x^{k_i} \overset{i\to\infty}{\longrightarrow} \left(A^TA\right)^{-1}A^T\left(b - By^\infty\right) = x^\infty.$$

Note that the optimality conditions (3.4) can be rewritten as

$$\begin{cases} \left(x - x^{k+1}\right)^T\left[Q^T\left(Qx^{k+1} - q\right) - A^T\left(\lambda^k - \beta\left(Ax^{k+1} + By^k - b\right)\right) - Q^Te_k(\eta^{k+1})\right] \geq 0, \\ g(y) - g\left(y^{k+1}\right) + \left(y - y^{k+1}\right)^T\left(-B^T\lambda^k + \beta B^T\left(Ax^{k+1} + By^{k+1} - b\right)\right) \geq 0, \\ \left(\lambda - \lambda^{k+1}\right)^T\left(Ax^{k+1} + By^{k+1} - b + \frac{1}{\beta}\left(\lambda^{k+1} - \lambda^k\right)\right) \geq 0, \end{cases}$$

for all $w \in \Omega$. Using the notation in (3.1), for the k-th iterate, we have

$$g(y) - g\left(y^k\right) + \left(w - w^k\right)^T F(w^k) + \left(w - w^k\right)^T\begin{pmatrix} \beta A^TB\left(y^{k-1} - y^k\right) - Q^Te_{k-1}(\eta^k) \\ 0 \\ \frac{1}{\beta}\left(\lambda^k - \lambda^{k-1}\right) \end{pmatrix} \geq 0, \tag{3.30}$$

for all $w \in \Omega$. Recall the results in Theorem 3.4. If we consider the subsequence $\{w^{k_i}\}$ converging to $w^\infty := (x^\infty, y^\infty, \lambda^\infty)$ and taking the limit in (3.30) over k_i, we obtain

$$g(y) - g\left(y^\infty\right) + \left(w - w^\infty\right)^T F(w^\infty) \geq 0, \quad \forall w \in \Omega,$$

which implies $w^\infty \in \Omega^\star$. Recall the convergence of all the following sequences:

$$\left\{u^{k_i} = \left(y^{k_i}, \lambda^{k_i}\right)\right\}, \quad \left\{\left\|e_k(\eta^{k+1})\right\|_2\right\} \quad \text{and} \quad \left\{\left\|u^k - u^{k+1}\right\|_{\mathcal{M}}\right\}.$$

As a result, for any $\varepsilon > 0$, there exists k_ℓ such that

$$\left\|u^{k_\ell} - u^\infty\right\|_{\mathcal{M}} < \varepsilon, \quad \left\|e_{k_\ell}(\eta^{k_\ell+1})\right\|_2 < \varepsilon \quad \text{and} \quad \left\|u^{k_\ell} - u^{k_\ell+1}\right\|_{\mathcal{M}} < \varepsilon.$$

Moreover, using the same technique for proving the inequality (3.23), for all $k > k_\ell + 1$, we have

$$\begin{aligned} \left\|u^k - u^\infty\right\|_{\mathcal{M}}^2 &\leq \left\|u^{k_\ell} - u^\infty\right\|_{\mathcal{M}}^2 + \frac{1}{\mu}\frac{\sigma}{1-\sigma}\left(\left\|e_{k_\ell}\left(\eta^{k_\ell+1}\right)\right\|_2 + \gamma\left\|u^{k_\ell} - u^{k_\ell+1}\right\|_{\mathcal{M}}\right)^2 \\ &\leq \varepsilon^2 + \frac{1}{\mu}\frac{\sigma}{1-\sigma}\left(\varepsilon + \gamma\varepsilon\right)^2, \end{aligned} \tag{3.31}$$

which directly implies $y^k \overset{k\to\infty}{\longrightarrow} y^\infty$ and $\lambda^k \overset{k\to\infty}{\longrightarrow} \lambda^\infty$. Recall the definition of x^k in (3.29) and the result $y^k \overset{k\to\infty}{\longrightarrow} y^\infty$, we further obtain the convergence of $\left\{x^k\right\}$ to x^∞. This completes the proof. □

Theorems 3.4 and 3.5 show that the proposed InADMM completely inherits the known convergence results of the original exact version of the ADMM (1.4). That is, we prove the convergence of the InADMM in sense of the constraint violation and optimal objective function value without the full-column-rank assumption of B; and its convergence in sense of iterates with this assumption. Indeed, without the full-column-rank assumption of B, the convergence result in Theorem 3.4 can be alternatively enhanced if the function $g(y)$ is assumed to be level bound. We summarize this extension in the following corollary.

Corollary 3.6. *Let $\{w^k\}$ be the sequence generated by the InADMM. If $g(y)$ is further assumed to be level bounded in (1.1), then the convergence of InADMM is partially in sense of iterates, i.e.,*

$$\left\{x^k\right\} \overset{k\to\infty}{\longrightarrow} x^\infty, \quad \left\{By^k\right\} \overset{k\to\infty}{\longrightarrow} By^\infty, \quad \left\{\lambda^k\right\} \overset{k\to\infty}{\longrightarrow} \lambda^\infty,$$

where $w^\infty = (x^\infty, y^\infty, \lambda^\infty) \in \Omega^\star$.

Proof. The proof is analogous to that of Theorem 3.5. First, recall the results in Theorem 3.4:

$$f\left(x^k\right) + g\left(y^k\right) \overset{k\to\infty}{\longrightarrow} f\left(x^\star\right) + g\left(y^\star\right). \tag{3.32}$$

Then, it follows from (3.24) that $Qx^k \overset{k\to\infty}{\longrightarrow} Qx^\star$ and thus $f\left(x^k\right) \overset{k\to\infty}{\longrightarrow} f\left(x^\star\right)$. Together with (3.32), we have $g\left(y^k\right) \overset{k\to\infty}{\longrightarrow} g\left(y^\star\right)$. If $g(y)$ is assumed to be level bounded, so is $\left\{y^k\right\}_{i=1}^\infty$ (see [38, Definition 1.8 in Chapter 1]). As a result, the sequence $\left\{u^k\right\}$ is bounded and the inequality (3.31) still holds. Since B is not necessarily full column rank, from (3.31) we only conclude that $By^k \overset{k\to\infty}{\longrightarrow} By^\infty$ and $\lambda^k \overset{k\to\infty}{\longrightarrow} \lambda^\infty$. The convergence of $\left\{x^k\right\}$ can be trivially derived by a similar proof as that of Theorem 3.5. The proof is complete. □

Remark 3.7. *For many concrete applications of the model (1.1), $g(y)$ is proper and level bounded. For examples, $g(y) = \|y\|_1$ in the LASSO model (1.6), the indicator function of a bounded closed set and the nuclear norm function (see,e.g. [37]), and any strongly convex function.*

Remark 3.8. *For our exclusive emphasis, the LASSO model (1.2) and its variant in distributed optimization (see (6.3) to be studied in Section 6.2), the function $g(y)$ is level bounded and the matrix B is full column rank. Therefore, the convergence in sense of iterates established in Theorem 3.5 is guaranteed for the sequence generated by the InADMM.*

4 Worst-case convergence rate

In [24, 25, 32], the $\mathcal{O}\left(\frac{1}{t}\right)$ worst-case convergence rate measured by the iteration complexity has been established for the exact version of ADMM (1.4) in both the ergodic and nonergodic senses, where t is the iteration counter. It this section, we extend similar analysis to the proposed InADMM. Despite of the similar roadmap in proofs, because of the consideration of the inexact solution for the subproblem (2.1), the analysis for InADMM turns out to be technically more complicated.

4.1 Ergodic convergence rate

We first prove the $\mathcal{O}\left(\frac{1}{t}\right)$ worst-case convergence rate in the ergodic sense for the InADMM. The proof for the exact version of the ADMM (1.4) is referred to [24, 32].

Theorem 4.1. *Let $\{w^k\}$ be the sequence generated by the proposed InADMM; and $\bar{w}^k$ be defined in (3.11). For any integer $t > 1$, we further define*

$$\hat{w}_t = \frac{1}{t}\sum_{k=1}^{t} \bar{w}^k. \tag{4.1}$$

Then, for all $w \in \Omega$, it holds that

$$g(\hat{y}_t) - g(y) + (\hat{w}_t - w)^T F(w) \leq \frac{1}{t}\left[\frac{1}{2\mu}\frac{\sigma}{1-\sigma}\left(\left\|e_0(\eta^1)\right\|_2 + \gamma\left\|u^0 - u^1\right\|_{\mathcal{M}}\right)^2 + \frac{1}{2}\left\|u^0 - u\right\|_{\mathcal{M}}^2\right] = \mathcal{O}\left(\frac{1}{t}\right), \tag{4.2}$$

where $\mathcal{M}$ is defined in (3.1), γ is defined in (3.9) and μ satisfies (3.10).

Proof. Recall the inequality (3.22). We thus have

$$\begin{aligned}
&\sum_{k=1}^{t}\left\{g(\bar{y}^k) - g(y) + \left(\bar{w}^k - w\right)^T F(w)\right\} \\
\leq\ & \frac{1}{2}\left(\left\|u^1 - u\right\|_{\mathcal{M}}^2 - \left\|u^{t+1} - u\right\|_{\mathcal{M}}^2\right) + \sum_{k=1}^{t}\left(\frac{\mu}{2}\frac{\sigma}{1-\sigma} - 1\right)\left\|Q\left(x^{k+1} - x\right)\right\|_2^2 \\
&+ \sum_{k=1}^{t-1}\frac{1}{2}\left(\frac{\sigma}{1-\sigma}\frac{\gamma^2}{\mu} - 1\right)\left\|u^k - u^{k+1}\right\|_{\mathcal{M}}^2 - \frac{1}{2}\left\|u^t - u^{t+1}\right\|_{\mathcal{M}}^2 + \frac{1}{2\mu}\frac{\sigma}{1-\sigma}\left(\left\|e_0(\eta^1)\right\|_2 + \gamma\left\|u^0 - u^1\right\|_{\mathcal{M}}\right)^2.
\end{aligned}$$

Then, using (3.10), for all $w \in \Omega$, we have

$$\begin{aligned}
& g(\hat{y}_t) - g(y) + (\hat{w}_t - w)^T F(w) \\
\overset{\text{Convexity}}{\leq}\ & \frac{1}{t}\sum_{k=1}^{t}\left\{g\left(\bar{y}^k\right) - g(y) + \left(\bar{w}^k - w\right)^T F(w)\right\} \\
\leq\ & \frac{1}{t}\left[\frac{1}{2\mu}\frac{\sigma}{1-\sigma}\left(\left\|e_0(\eta^1)\right\|_2 + \gamma\left\|u^0 - u^1\right\|_{\mathcal{M}}\right)^2 + \frac{1}{2}\left\|u^1 - u\right\|_{\mathcal{M}}^2\right],
\end{aligned}$$

which completes the proof. □

Theorem 4.1 shows that after t iterations of the InADMM, we can find an approximate solution of the variational inequality (3.2) with an accuracy of $\mathcal{O}\left(\frac{1}{t}\right)$. This approximate solution is given in (4.1) and it is the average of all the points $\bar{w}^k$ which can be computed by all the known iterates generated by the InADMM. Hence, this is an $\mathcal{O}\left(\frac{1}{t}\right)$ worst-case convergence rate in the ergodic sense for the proposed InADMM.

4.2 Non-ergodic convergence rate

Then we prove the $\mathcal{O}\left(\frac{1}{t}\right)$ worst-case convergence rate in a non-ergodic sense. Note that the proof for the exact version of the ADMM (1.4) is referred to [25].

To estimate the worst-case convergence rate in a non-ergodic sense, we first need to clarify a criterion to precisely measure the accuracy of an iterate. Recall the optimality condition of an iterate generated by the InADMM is given in (3.4). Then, it is easy to derive that for the iterate $(x^{k+1}, y^{k+1}, \lambda^{k+1})$ generated by the InADMM, for all $w \in \Omega$, it holds that

$$g(y) - g(y^{k+1}) + \left(w - w^{k+1}\right)^T F(w^{k+1}) + \left(w - w^{k+1}\right)^T \begin{pmatrix} \beta A^T B\left(y^k - y^{k+1}\right) - Q^T e_k(\eta^{k+1}) \\ 0 \\ \frac{1}{\beta}\left(\lambda^{k+1} - \lambda^k\right) \end{pmatrix} \geq 0.$$

Recall the variational inequality reformulation (3.2) and the notation in (3.1). It is clear that $(x^{k+1}, y^{k+1}, \lambda^{k+1})$ is a solution point of (3.2) if and only if $\left\|u^k - u^{k+1}\right\|_{\mathcal{M}}^2 = 0$ and $\left\|e_k(\eta^{k+1})\right\|_2^2 = 0$. Hence, it is reasonable to measure the accuracy of the iterate $(x^{k+1}, y^{k+1}, \lambda^{k+1})$ by $\left\|u^k - u^{k+1}\right\|_{\mathcal{M}}^2$ and $\left\|e_k(\eta^{k+1})\right\|_2^2$. Our purpose is thus to prove that after t iterations of the InADMM, both $\left\|u^k - u^{k+1}\right\|_{\mathcal{M}}^2$ and $\left\|e_k(\eta^{k+1})\right\|_2^2$ can be bounded by upper bounds in order of $\mathcal{O}\left(\frac{1}{t}\right)$.

Theorem 4.2. *Let $\{w^k\}$ be the sequence generated by the InADMM. Then, for any integer $t > 1$, we have*

$$\min_{1\le k\le t}\left\{\left\|u^k - u^{k+1}\right\|_{\mathcal{M}}^2\right\} \le \frac{1}{t}\left[\frac{1}{\nu}\left\|u^1 - u^\star\right\|_{\mathcal{M}}^2 + \frac{1}{\nu\mu}\frac{\sigma}{1-\sigma}\left(\left\|e_0(\eta^1)\right\|_2 + \gamma\left\|u^0 - u^1\right\|_{\mathcal{M}}\right)^2\right] \tag{4.3}$$

and

$$\begin{aligned}\min_{1\le k\le t}\left\{\left\|e_k(\eta^{k+1})\right\|_2^2\right\} \le \frac{1}{t^2}&\left[2\left(\frac{\sigma}{1-\sigma}\right)^2\cdot\left(\left\|e_0(\eta^1)\right\|_2 + \gamma\left\|u^0 - u^1\right\|_{\mathcal{M}}\right)^2\right]\\ &+\frac{1}{t}\left\{2\left(\frac{\sigma}{1-\sigma}\gamma\right)^2\cdot\left[\frac{1}{\nu}\left\|u^1 - u^\star\right\|_{\mathcal{M}}^2 + \frac{1}{\nu\mu}\frac{\sigma}{1-\sigma}\left(\left\|e_0(\eta^1)\right\|_2 + \gamma\left\|u^0 - u^1\right\|_{\mathcal{M}}\right)^2\right]\right\},\end{aligned} \tag{4.4}$$

where $w^\star \in \Omega^\star$, γ is defined as (3.9), μ satisfies (3.10) and $\nu = 1 - \frac{\gamma^2}{\mu}\frac{\sigma}{1-\sigma} > 0$.

Proof. Recall the inequality (3.23) and choose $w^\star \in \Omega^\star$. We obtain

$$\begin{aligned}&\sum_{k=1}^{t+1}\left(1 - \frac{\mu}{2}\frac{\sigma}{1-\sigma}\right)\left\|Q\left(x^{k+1} - x^\star\right)\right\|_2^2 + \sum_{k=1}^{t}\left(\frac{1}{2} - \frac{\gamma^2}{2\mu}\frac{\sigma}{1-\sigma}\right)\left\|u^k - u^{k+1}\right\|_{\mathcal{M}}^2\\ \le\ &\frac{1}{2}\left\|u^1 - u^\star\right\|_{\mathcal{M}}^2 + \frac{1}{2\mu}\frac{\sigma}{1-\sigma}\left(\left\|e_0(\eta^1)\right\|_2 + \gamma\left\|u^0 - u^1\right\|_{\mathcal{M}}\right)^2 - \frac{1}{2}\left\|u^{t+2} - u^\star\right\|_{\mathcal{M}}^2,\end{aligned} \tag{4.5}$$

which implies

$$\sum_{k=1}^{t}\left(\frac{1}{2} - \frac{\gamma^2}{2\mu}\frac{\sigma}{1-\sigma}\right)\left\|u^k - u^{k+1}\right\|_{\mathcal{M}}^2 \le \frac{1}{2}\left\|u^1 - u^\star\right\|_{\mathcal{M}}^2 + \frac{1}{2\mu}\frac{\sigma}{1-\sigma}\left(\left\|e_0(\eta^1)\right\|_2 + \gamma\left\|u^0 - u^1\right\|_{\mathcal{M}}\right)^2. \tag{4.6}$$

Consequently, we have

$$\min_{1\le k\le t}\left\{\left\|u^k - u^{k+1}\right\|_{\mathcal{M}}^2\right\} \le \frac{1}{t}\left[\frac{1}{\nu}\left\|u^1 - u^\star\right\|_{\mathcal{M}}^2 + \frac{1}{\nu\mu}\frac{\sigma}{1-\sigma}\left(\left\|e_0(\eta^1)\right\|_2 + \gamma\left\|u^0 - u^1\right\|_{\mathcal{M}}\right)^2\right],$$

and the assertion (4.3) is proved. Note that ν is positive because of (3.10).

Then, it follows from the inequality (3.18) that

$$\left\|e_k(\eta^{k+1})\right\|_2 \le \sum_{i=0}^{k-1}\sigma^{k-i}\gamma\left\|u^i - u^{i+1}\right\|_{\mathcal{M}} + \sigma^k\left\|e_0(\eta^1)\right\|_2.$$

Summarizing the above inequality from $k = 1$ to $k = t$, we obtain

$$\sum_{k=1}^{t}\left\|e_k(\eta^{k+1})\right\|_2 \le \sum_{k=1}^{t}\left\{\sum_{i=0}^{k-1}\sigma^{k-i}\gamma\left\|u^i - u^{i+1}\right\|_{\mathcal{M}} + \sigma^k\left\|e_0(\eta^1)\right\|_2\right\}. \tag{4.7}$$

In addition, we have

$$\begin{aligned}\sum_{k=1}^{t}\sum_{i=0}^{k-1}\sigma^{k-i}\gamma\cdot\left\|u^i-u^{i+1}\right\|_{\mathcal{M}} &= \sum_{i=0}^{t-1}\sum_{k=i+1}^{t}\sigma^{k-i}\gamma\cdot\left\|u^i-u^{i+1}\right\|_{\mathcal{M}}\\ &= \sum_{i=0}^{t-1}\frac{\sigma-\sigma^{t-i+1}}{1-\sigma}\gamma\cdot\left\|u^i-u^{i+1}\right\|_{\mathcal{M}}=\sum_{i=0}^{t-1}\frac{\sigma}{1-\sigma}\gamma\cdot\left\|u^i-u^{i+1}\right\|_{\mathcal{M}},\end{aligned}$$

and

$$\sum_{k=1}^{t}\sigma^k\left\|e_0(\eta^1)\right\|_2\le\frac{\sigma-\sigma^{t+1}}{1-\sigma}\left\|e_0(\eta^1)\right\|_2\le\frac{\sigma}{1-\sigma}\left\|e_0(\eta^1)\right\|_2.$$

Then, by simple calculation, we have

$$\begin{aligned}&\left(\sum_{k=1}^{t}\left\|e_k(\eta^{k+1})\right\|_2\right)^2\\ \overset{(4.7)}{\le}\ &\left(\sum_{i=0}^{t-1}\frac{\sigma}{1-\sigma}\gamma\cdot\left\|u^i-u^{i+1}\right\|_{\mathcal{M}}+\frac{\sigma}{1-\sigma}\left\|e_0(\eta^1)\right\|_2\right)^2\\ \le\ &2\left(\sum_{i=1}^{t-1}\frac{\sigma}{1-\sigma}\gamma\cdot\left\|u^i-u^{i+1}\right\|_{\mathcal{M}}\right)^2+2\left(\frac{\sigma}{1-\sigma}\cdot\left(\left\|e_0(\eta^1)\right\|_2+\gamma\left\|u^0-u^1\right\|_{\mathcal{M}}\right)\right)^2\\ \le\ &2\left(\frac{\sigma}{1-\sigma}\gamma\right)^2\cdot\left(\sum_{i=1}^{t}\left\|u^i-u^{i+1}\right\|_{\mathcal{M}}\right)^2+2\left(\frac{\sigma}{1-\sigma}\right)^2\cdot\left(\left\|e_0(\eta^1)\right\|_2+\gamma\left\|u^0-u^1\right\|_{\mathcal{M}}\right)^2\\ \le\ &2\left(\frac{\sigma}{1-\sigma}\gamma\right)^2 t\cdot\left(\sum_{i=1}^{t}\left\|u^i-u^{i+1}\right\|_{\mathcal{M}}^2\right)+2\left(\frac{\sigma}{1-\sigma}\right)^2\cdot\left(\left\|e_0(\eta^1)\right\|_2+\gamma\left\|u^0-u^1\right\|_{\mathcal{M}}\right)^2\\ \overset{(4.6)}{\le}\ &2\left(\frac{\sigma}{1-\sigma}\gamma\right)^2 t\cdot\left[\frac{1}{\nu}\left\|u^1-u^\star\right\|_{\mathcal{M}}^2+\frac{1}{\nu\mu}\frac{\sigma}{1-\sigma}\left(\left\|e_0(\eta^1)\right\|_2+\gamma\left\|u^0-u^1\right\|_{\mathcal{M}}\right)^2\right]\\ &+2\left(\frac{\sigma}{1-\sigma}\right)^2\cdot\left(\left\|e_0(\eta^1)\right\|_2+\gamma\left\|u^0-u^1\right\|_{\mathcal{M}}\right)^2.\end{aligned}$$

Further we get

$$\begin{aligned}&\left(t\cdot\min_{1\le k\le t}\left\|e_k\left(\eta^{k+1}\right)\right\|_2\right)^2\le\left(\sum_{k=1}^{t}\left\|e_k(\eta^{k+1})\right\|_2\right)^2\\ \le\ &2\left(\frac{\sigma}{1-\sigma}\gamma\right)^2 t\cdot\left[\frac{1}{\nu}\left\|u^1-u^\star\right\|_{\mathcal{M}}^2+\frac{1}{\nu\mu}\frac{\sigma}{1-\sigma}\left(\left\|e_0(\eta^1)\right\|_2+\gamma\left\|u^0-u^1\right\|_{\mathcal{M}}\right)^2\right]\\ &+2\left(\frac{\sigma}{1-\sigma}\right)^2\cdot\left(\left\|e_0(\eta^1)\right\|_2+\gamma\left\|u^0-u^1\right\|_{\mathcal{M}}\right)^2,\end{aligned}\tag{4.8}$$

which implies

$$\begin{aligned}&\min_{1\le k\le t}\left\{\left\|e_k(\eta^{k+1})\right\|_2^2\right\}\\ \le\ &\frac{1}{t}\left\{2\left(\frac{\sigma}{1-\sigma}\gamma\right)^2\cdot\left[\frac{1}{\nu}\left\|u^1-u^\star\right\|_{\mathcal{M}}^2+\frac{1}{\nu\mu}\frac{\sigma}{1-\sigma}\left(\left\|e_0(\eta^1)\right\|_2+\gamma\left\|u^0-u^1\right\|_{\mathcal{M}}\right)^2\right]\right\}\\ &+\frac{1}{t^2}\left[2\left(\frac{\sigma}{1-\sigma}\right)^2\cdot\left(\left\|e_0(\eta^1)\right\|_2+\gamma\left\|u^0-u^1\right\|_{\mathcal{M}}\right)^2\right].\end{aligned}$$

The proof is complete. □

Note that the numbers in the right-hand sides of both (4.3) and (4.4) are order of $\mathcal{O}\left(\frac{1}{t}\right)$. Hence, Theorem 4.2 offers the $\mathcal{O}\left(\frac{1}{t}\right)$ worst-case convergence rate in a non-ergodic sense for the proposed InADMM.

In this section, we show that the proposed InADMM also inherits the known worst-case convergence rates established in [24, 25, 32] for the original exact version of the ADMM (1.4).

5 Safe-guard numbers for internally nested iterations

In this section, we discuss how to ensure the inexactness criterion (2.9) when a standard numerical linear algebra solver is applied to iteratively solve the linear system (2.5a).

Recall that (2.5a) is

$$\hat{H}\eta^{k+1} - Q\left(\beta A^T A\right)^{-1} h^k = 0. \tag{5.1}$$

We aim at finding the safe-guard iteration number, denoted by $n_{\max}(\sigma)$, for a specific solver if it is applied to iteratively solve the linear system (5.1), so as to meet the inexactness criterion (2.9). Hence, the implementation of InADMM is automatic without any ambiguity and user-friendly. Note that n is the counter for the internally nest iterations and the counter k for the outer-loop iterations is fixed in this section.

Let η_n^k denote the n-th internal iterate generated by a solver for the linear system (5.1) and set $\eta_0^k = \eta^k$ as the initial iterate. The matrix $\hat{H}$ defined in (2.6) is positive definite because of the full-column-rank assumption of A. So we use the matrix norm $\|\eta\|_{\hat{H}} := \sqrt{\eta^T \hat{H} \eta}$. Let us denote the spectrum radius of $\hat{H}$ by $\rho(\hat{H})$, and its condition number by $\kappa(\hat{H})$.

5.1 CG and PCG

First, we analyze the safe-guard number for the CG. According to, e.g. [40, Lecture 38, Theorem 38.5], the error at the n-th iterate generated by the CG satisfies

$$\left\|\eta_n^k - \eta_{exact}^k\right\|_{\hat{H}} \leq 2c^n \left\|\eta_0^k - \eta_{exact}^k\right\|_{\hat{H}}, \tag{5.2}$$

where η_{exact}^k denotes the exact solution of the linear system (5.1), and the constant c is defined as

$$c = \frac{\sqrt{\kappa(\hat{H})} - 1}{\sqrt{\kappa(\hat{H})} + 1} < 1.$$

Therefore, for the n-th iterate η_n^k generated by the CG, it satisfies that

$$\begin{aligned}
\left\|e_k\left(\eta_n^k\right)\right\|_2 &= \left\|\hat{H}\eta_n^k - Q\left(\beta A^T A\right)^{-1} h^k\right\|_2 \\
&\leq \sqrt{\rho(\hat{H})} \cdot \left\|\eta_n^k - \eta_{exact}^k\right\|_{\hat{H}} \overset{(5.2)}{\leq} 2\sqrt{\rho(\hat{H})} \cdot c^n \left\|\eta_0^k - \eta_{exact}^k\right\|_{\hat{H}} \\
&\leq 2\sqrt{\kappa(\hat{H})} \cdot c^n \left\|\hat{H}\left(\eta^k - \eta_{exact}^k\right)\right\|_2 = 2\sqrt{\kappa(\hat{H})} \cdot c^n \left\|\hat{H}\eta^k - Q\left(\beta A^T A\right)^{-1} h^k\right\|_2, \\
&= 2\sqrt{\kappa(\hat{H})} \cdot c^n \left\|e_k\left(\eta^k\right)\right\|_2,
\end{aligned}$$

where $e_k(\eta)$ is defined in (2.8) and note that $\eta_0^k = \eta^k$. Obviously, to guarantee the inexactness criterion (2.9) with a given σ, it suffices to hold

$$2\sqrt{\kappa(\hat{H})} \cdot c^{n(\sigma)} \leq \sigma.$$

In other words, to apply the CG for iteratively solving (5.1), the inexactness criterion (2.9) is guaranteed by the safe-guard iteration number:

$$n_{\max}(\sigma) := \left\lceil \log_c \left(\sigma / \left(2\sqrt{\kappa(\hat{H})} \right) \right) \right\rceil. \tag{5.3}$$

In addition, if the linear system (5.1) is ill conditioned, *i.e.*, $\kappa(\hat{H})$ is large, we consider using the PCG for the linear system (5.1). For this case, the preconditioned surrogate of (5.1) is solved instead:

$$P^{-1} \left(\hat{H}\eta^{k+1} - Q \left(\beta A^T A \right)^{-1} h^k \right) = 0,$$

in which P is the preconditioner. For this case, the safe-guard iteration number for the PCG is given by (5.3) but with $c = \frac{\sqrt{\kappa(P^{-1}\hat{H})}-1}{\sqrt{\kappa(P^{-1}\hat{H})}+1}$.

5.2 Jacobian, Gauss-Seidel and SOR

For other solvers such as the Jacobian, Gauss-Seidel and SOR methods, similar analysis can be conducted for finding the safe-guard numbers. More specifically, we decompose the matrix $\hat{H}$ as

$$\hat{H} := D - L^T - L,$$

where D is a diagonal matrix and L is a lower triangular matrix. Let us consider a conceptual and general iterative scheme

$$\eta_{n+1}^k = T\eta_n^k + S^{-1}h^k, \tag{5.4}$$

where S and T satisfy $S - ST = \hat{H}$ with $\rho(T) < 1$ to ensure the convergence. Then, the Jacobian, Gauss-Seidel and SOR methods can all be specified by the general scheme (5.4) as follows

$$\begin{cases} \textbf{Jacobi:} & T = D^{-1}(L^T + L), \quad S = D, \\ \textbf{Gauss-Seidel:} & T = (D-L)^{-1}L^T, \quad S = (D-L), \\ \textbf{SOR:} & T = (D - wL)^{-1}((1-w)D + wL^T), \quad S = (\frac{D}{w} - L). \end{cases} \tag{5.5}$$

Our analysis is thus valid for any scheme that can be recovered by the general scheme (5.4), though the mentioned three ones are still our main purpose.

According to [41, Chapter 3.2], for the n-th iteration generated by the scheme (5.4) for (5.1), η_n^k, it holds that

$$\begin{aligned} \left\| e_k\left(\eta_n^k\right) \right\|_2 &= \left\| \hat{H}\eta_n^k - Q\left(\beta A^T A\right)^{-1} h^k \right\|_2 = \left\| \hat{H}\left(\eta_n^k - \eta_{exact}^k\right) \right\|_2 = \left\| \hat{H}T^n \left(\eta_0^k - \eta_{exact}^k\right) \right\|_2 \\ &\le \left\| \hat{H}T^n\hat{H}^{-1} \right\|_2 \cdot \left\| \hat{H}\left(\eta_0^k - \eta_{exact}^k\right) \right\|_2 = \left\| \hat{H}T^n\hat{H}^{-1} \right\|_2 \cdot \left\| \hat{H}\eta_n^k - Q\left(\beta A^T A\right)^{-1} h^k \right\|_2 \\ &= \left\| \hat{H}T^n\hat{H}^{-1} \right\|_2 \cdot \left\| e_k\left(\eta_0^k\right) \right\|_2. \end{aligned} \tag{5.6}$$

Therefore, to ensure the inexactness criterion (2.9) at η_n^k, it suffices to guarantee $\left\| \hat{H}T^n\hat{H}^{-1} \right\|_2 \le \sigma$. Since $\rho(T) < 1$ is required to ensure the convergence of the general scheme (5.4), immediately we know that $\left\| \hat{H}T^n\hat{H}^{-1} \right\|_2 \overset{n\to\infty}{\longrightarrow} 0$. Hence, $\left\| \hat{H}T^n\hat{H}^{-1} \right\|_2$ must be smaller than the given positive scalar σ for a sufficiently large n and the safe-guard number $n_{\max}(\sigma)$ can be discerned accordingly on the cost of estimating $\left\| \hat{H}T^n\hat{H}^{-1} \right\|_2$. Below we give some more meticulous analysis for estimating the safe-guard numbers for (5.4).

Let us recall the average rate of convergence and asymptotic rate of convergence, see, e.g., [41, Theorem 3.4 Chapter 3]. Then we have

$$\begin{aligned}\lim_{n\to\infty}\frac{-\ln\left\|\hat{H}T^n\hat{H}^{-1}\right\|_2}{n} &= \lim_{n\to\infty}\frac{-\ln\left\|(\hat{H}T\hat{H}^{-1})^n\right\|_2}{n}\\ &= -\ln\rho(\hat{H}T\hat{H}^{-1}) := R_\infty(\hat{H}T\hat{H}^{-1})\\ &= -\ln\rho(T). \end{aligned}\tag{5.7}$$

Hence, instead of estimating $\left\|\hat{H}T^n\hat{H}^{-1}\right\|_2$ which is usually computationally expensive, we can replace it by $[\rho(T)]^n$ and accordingly estimate the safe-guard number via $[\rho(T)]^n < \sigma$. That is, the safe-guard number for (5.4) can be well estimated by the number $\left\lceil \log_{\rho(T)}\sigma \right\rceil$. For some specific cases of the general scheme (5.4), this estimate can be precise. For example, if $S = aI$ with $a > \rho(\hat{H})$ in (5.4), then we have $\left\|\hat{H}T^n\hat{H}^{-1}\right\|_2 = \rho^n(T)$ and thus the safe-guard number is precisely given by $n_{\max}(\sigma) = \left\lceil \log_{\rho(T)}\sigma \right\rceil$. Also, if it is known that the condition $\left\|\hat{H}T\hat{H}^{-1}\right\|_2 < 1$ is satisfied, then we have

$$\left\|\hat{H}T^n\hat{H}^{-1}\right\|_2 = \left\|(\hat{H}T\hat{H}^{-1})^n\right\|_2 \le \left\|\hat{H}T\hat{H}^{-1}\right\|_2^n,$$

with which the safe-guard number is precisely given by $n_{\max}(\sigma) = \left\lceil \log_{\left\|\hat{H}T\hat{H}^{-1}\right\|_2}\sigma \right\rceil$.

As we shall show in Section 6, usually these safe-guard numbers are very small, in one or two digits, despite that the dimension of the involved linear system is huge. Hence, the internally nested iterations to meet the inexactness criterion (2.9) via iteratively solving the linear system (5.1) can be very efficient. This feature guarantees the efficiency of the proposed InADMM for big-data scenarios of (1.1) with huge-dimensional variables. Last, we reiterate that the safe-guard numbers are sufficient to guarantee the inexact criterion (2.9) and they are still over-estimated; we refer to numerical results in Section 6. To implement the proposed InADMM, the inexactness criterion (2.9) automatically guarantees the safe level of accuracy and there is no need to follow these safe-guard numbers to execute the internally nested iterations.

6 Numerical experiments

In this section we test some big datasets of the benchmark LASSO model (1.2) and its variant arising in distributed optimization (see (6.3)); and numerically show the efficiency of the proposed InADMM. Our theoretical assertions, especially the necessity of solving the involved systems of linear equations inexactly subject to the proposed inexactness criterion for big datasets, are numerically verified. All the numerical experiments were implemented on a Laptop with Intel(R) Core(TM) i5-6300U CPU@ 2.40GHz 2.50GHz and 8.00 GB Memory. All the codes were written in MATLAB2016A.

6.1 LASSO

Recall that the benchmark LASSO model (1.2) can be reformulated as (1.6) and the iterative scheme of the exact version of ADMM reads as (1.7). To apply the proposed InADMM, the resulting y- and λ-subproblems are the same as (1.7) except that the x-subproblem (1.8) is solved inexactly by (2.5a) subject to the inexactness criterion (2.9)-(2.10).

6.1.1 Synthetic dataset

We first generate some synthetic datasets for the benchmark LASSO model (1.2) with gradually increasing dimensionality.

The matrix Q is generated by the Matlab script "*sprandn(p,n,d)*" where p and n are the dimensions and d is the density of nonzero entries. We report 5 cases as shown in Table 1. Then, a vector $x_0 \in \mathbb{R}^n$ is generated by "*sprandn(n,1,$\frac{100}{n}$)*" and the vector $q \in \mathbb{R}^p$ is calculated by $Qx_0 + 0.1 \cdot \varepsilon$ with $\varepsilon \in \mathbb{R}^p$ a standard normally distributed random noise vector. Following [5], the parameter τ is set to be $0.1 \cdot \|Q^T q\|_\infty$ for effectively identifying nonzero entries.

To illustrate the necessity of solving the x-subproblem (1.8) inexactly, we also test the case where these subproblems are solved exactly by direct methods when the dimension of the linear system (1.8) is not that high. This is also the reason why we purposively generate some small- or medium-size synthetic datasets so that the x-subproblem can be solved exactly by direct methods and thus the exact and inexact versions of ADMM can be compared. We particularly test Cholesky factorization and the LSQR method as in [5][2]. The corresponding iterative schemes are denoted by ADMM$_{Cholesky}$ and ADMM$_{LSQR}$, respectively. That is, ADMM$_{Cholesky}$ means the Cholesky factorization $\hat{H} = LL^T$ is executed and then the solution of (2.5a) is given by $\eta^{k+1} = \frac{1}{\beta} L^{-T} \left(L^{-1} Q h^k \right)$. As in [5], the stopping criterion for implementing ADMM$_{Cholesky}$ and ADMM$_{LSQR}$ is:

$$\left\| x^k - y^k \right\|_2 < \sqrt{n} \times \epsilon^{abs} + \epsilon^{rel} \times \max \left\{ \left\| x^k \right\|_2, \left\| y^k \right\|_2 \right\},$$

$$\left\| \beta \left(y^k - y^{k-1} \right) \right\|_2 < \sqrt{n} \times \epsilon^{abs} + \epsilon^{rel} \times \left\| \lambda^k \right\|_2,$$

where $\epsilon^{abs} = 10^{-4}$ and $\epsilon^{rel} = 10^{-3}$. Certainly, if too huge-dimensional cases are considered, it is hard to apply these direct methods to solve the x-subproblem (1.8) exactly.

Note that when the LASSO model (1.2) is considered, we have $A^T A = I_{n \times n}$ and hence the linear system (2.5a) reduces to

$$\hat{H} \eta = \frac{1}{\beta} Q h^k \tag{6.1}$$

with $\hat{H}$ and h^k given in (2.6) and (2.3), respectively.

Meanwhile, to show the superiority of the proposed automatically adjustable inexactness criterion (2.9), we particularly compare it with the cases where the linear system (6.1) is solved inexactly but subject to a-prior fixed levels of accuracy:

$$\frac{\left\| \frac{1}{\beta} Q h^k - \hat{H} \eta \right\|_2}{\left\| \frac{1}{\beta} Q h^k \right\|_2} \leq 10^{-t},$$

where the integer t denotes an accuracy level. We shall test various values of $t = 2, 4, 6, 8, 10$, representing low to high fixed levels of accuracy.

For simplicity, let us just focus on implementing the CG for the linear system (6.1). According to Section 5, the safe-guard iteration number for the CG is given by (5.3). In practice, if it is computationally expensive to compute $\kappa(\hat{H})$ exactly, then instead of (5.3), we can also compute a upper bound of κ as $\kappa_u := \frac{\|Q\|_2^2}{\beta} + 1$ and accordingly $c_u := \frac{\sqrt{\kappa_u} - 1}{\sqrt{\kappa_u} + 1}$ which is also a upper bound of c, and estimate the safe-guard iteration number $n_{\max}(\sigma)$ theoretically given in (5.3) via

$$\left\lceil \log_{c_u} \left(\sigma / \left(2\sqrt{\kappa_u} \right) \right) \right\rceil. \tag{6.2}$$

[2] http://stanford.edu/~boyd/papers/admm/

The initial iterate (η^0, y^0, λ^0) is set to be zero, $\beta = 0.05 \cdot \|Q^T q\|_\infty$ and σ is chosen as

$$\sigma = \frac{0.99}{1 + \frac{\|Q\|_2}{\sqrt{2\beta}}} < \frac{\sqrt{2\beta}}{\sqrt{2\beta} + \|Q\|_2},$$

so as to satisfy the condition (2.10) and $\|Q\|_2$ is obtained by the state-of-art power iteration in [31]. For comparison, all the inexact versions of the ADMM, including InADMM and the cases where the x-subproblem (1.8) is solved up to fixed accuracy levels, terminate when the objective function values are better than those obtained by the ADMM$_{Cholesky}$ and ADMM$_{LSQR}$. That is,

$$\frac{1}{2}\left\|Qy^k - q\right\|_2^2 + \tau\left\|y^k\right\|_1 < \min\left\{\text{Objective of ADMM}_{Cholesky}, \text{Objective of ADMM}_{LSQR}\right\},$$

which is reasonable if we recall that the LASSO model (1.2) is unconstrained. For all the methods under comparison, the initial iterate for executing the $(k+1)$-th's internally nested iteration is taken as the k-th outer-loop iterate η^k.

We list the parameters defining the synthetic datasets in Table 1, and the corresponding values of τ, β, σ and the safe-guard numbers of the CG. We report some numerical results in Table 2 for the synthetic datasets. In this table, "Iteration" means the overall outer iteration number, "Mean/Max CG" are the mean and maximal iteration numbers of the CG for solving the linear systems among all outer iterations, "Time" is the computing time in seconds, "Obj" is objective function value when the stopping criterion is satisfied, and "$n_{\max}(\sigma)$" is the safe-guard iteration number of the CG computed by (6.2) to guarantee the inexactness criterion (2.9) when the InADMM is implemented. We label "$\sim$" for the case where the iteration number exceeds the maximum outer-loop iteration number (which is set as 500 in our code) or where it is inapplicable.

Table 1: Values of τ, β, σ and $n\,(\sigma)$ for synthetic datasets

(p, n, d)	τ	β	σ	$n_{\max}\,(\sigma)$
$(10^4, 1.5*10^4, 50\%)$	1067.321	533.6606	0.1889	13
$(10^4, 1.5*10^4, 20\%)$	540.2915	270.1457	0.1960	12
$(2.5*10^4, 5*10^4, 1\%)$	73.1531	36.5766	0.1818	14
$(10^5, 1.5*10^5, 0.1\%)$	25.3148	12.6574	0.1816	15
$(10^5, 10^6, 0.01\%)$	4.0682	2.0341	0.1227	26

We see that both $ADMM_{Cholesky}$ and $ADMM_{LSQR}$ are generally much slower than inexact versions of the ADMM which solve this linear system inexactly, especially if the dimension of dataset is larger. In our experiments, the time for Cholesky decomposition exceeds 5000 seconds for the latter two cases and thus their comparisons are not included in Table 2. Results in this table show that generally it is necessary to solve the linear systems (6.1) inexactly when the dimension is high. Also, it is not efficient to control the accuracy of the inexact solutions of (6.1) by neither too low nor too high accuracy; and there is no evidence in fixing which level of the accuracy. With the automatically adjustable inexactness criterion (2.9), the proposed InADMM works well for all the tested cases and it automatically avoids the difficulty caused by extremely low or high accuracy for the linear system (6.1). It is conclusive that solving the linear system (6.1) up to a too high accuracy does not help at all in accelerating the convergence of the ADMM; meanwhile it is easy to imagine that solving it with a too low accuracy returns low-quality output and hence ruin the convergence.

Our experiments show that the accuracy of 10^{-4} turns out to be good for the generated datasets, but the point is that there is no clue for choosing the accuracy level a prior. As theoretically analyzed, despite of the high dimension of the linear system (6.1), only a few CG iterations are needed to satisfy the inexactness criterion (2.9) and hence to guarantee the convergence of InADMM. This feature significantly helps save computation for big-data cases of the LASSO model (1.2). We notice that $n_{\max}(\sigma)$ for InADMM is just a theoretical and overestimated upper bound to guarantee (2.9); practically the iteration numbers that are really executed by the CG for these datasets to ensure (2.9) are just at most 2 or 3. This essentially explains the efficiency of the InADMM shown in Table 2.

6.1.2 Real dataset

We also test two popular real datasets: "RCV1"[3] in [28] and "news20"[4] in [27] for the LASSO model (1.2). Implementation details of various versions of the ADMM are the same as those stated in the last subsection, without otherwise specified. The ADMM$_{Cholesky}$ is not compared in this subsection because the dimensions of the these two datasets are too large and it is too time-consuming to execute the Cholesky factorization.

For "RCV1", in terms of the LASSO model (1.2), the dimension of the data matrix Q is 20242×47236. We set $\tau = 0.1 \cdot \|Q^T q\|_\infty = 26.4381$, $\beta = 0.05 \cdot \|Q^T q\|_\infty = 13.2190$ and $\sigma = 0.1934$ with $n_{\max}(\sigma) = 13$. The matrix Q is sparse and thus generally all the tested versions of the ADMM are quite fast. We report some numerical results in Table 3, from which the efficiency of InADMM is clearly shown. For this dataset, it is empirically observed that the accuracy of 10^{-4} is good for the internally nested iterations. Also, for this dataset, the InADMM requires only 3 CG steps to meet the inexactness criterion (2.9) and hence it is very efficient.

Table 3: Numerical Comparison on "RCV1" Dataset

Algorithm	Iteration	Mean CG Number	Max CG Number	Time	Obj
ADMM$_{LSQR}$	21	$\sim$	$\sim$	2.5185	7.0631e+03
ADMM$_{1e-10}$	21	16.9524	20	4.0077	7.0631e+03
ADMM$_{1e-8}$	21	13.6190	17	3.2112	7.0631e+03
ADMM$_{1e-6}$	21	9.2857	12	2.3173	7.0631e+03
ADMM$_{1e-4}$	21	5.1429	8	1.4650	7.0631e+03
ADMM$_{1e-2}$	> 500	$\sim$	$\sim$	$\sim$	$\sim$
InADMM	22	2.8182	3	1.1996	7.0631e+03

We further test the "news20" dataset. In terms of the LASSO model (1.2), the dimension of the data matrix Q for this dataset is $19,996 \times 1,355,191$. Note that Q is also spare for this dataset. We take $\tau = 0.1\|Q^T q\|_\infty = 12.3306$, $\beta = 0.05\|Q^T q\|_\infty = 6.1653$, and $\sigma = 0.0921$ with $n_{\max}(\sigma) = 35$. Some numerical results are reported in Table 4. For this dataset, the accuracy of 10^{-4} is also good for the internally nested iterations and the InADMM requires 4 CG steps to meet the inexactness criterion (2.9). We further plot the evolutions of objective function values with respect to the computing time, and error of subproblems with respect to the outer-loop iterations in Figure 1. The curves in this figure show that InADMM achieves the near-optimal objective function value faster

[3] http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html♯rcv1.binary
[4] http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html♯news20

Table 2: Comparison between InADMM, $\text{ADMM}_{Cholesky}$, ADMM_{LSQR} and ADMM_{1e-t}

(p, n, d)	Algorithm	Iteration	Mean/Max CG	Time	Obj
$(10^4, 1.5*10^4, 50\%)$	$\text{ADMM}_{Cholesky}$	87	$\sim$	58.4204	6.0922e+04
	ADMM_{LSQR}	87	$\sim$	222.6887	6.0922e+04
	ADMM_{1e-10}	87	20.5172/29	459.7782	6.0922e+04
	ADMM_{1e-8}	87	14.1034/23	342.5131	6.0922e+04
	ADMM_{1e-6}	87	7.5862/16	193.0946	6.0922e+04
	ADMM_{1e-4}	82	2.6098/10	86.5464	6.0922e+04
	ADMM_{1e-2}	> 500	$\sim$	$\sim$	$\sim$
	InADMM	85	1.2471/2	69.3687	6.0922e+04
$(10^4, 1.5*10^4, 20\%)$	$\text{ADMM}_{Cholesky}$	80	$\sim$	57.1158	3.6288e+04
	ADMM_{LSQR}	80	$\sim$	96.9766	3.6288e+04
	ADMM_{1e-10}	80	19.9500/29	198.0422	3.6288e+04
	ADMM_{1e-8}	80	13.8125/23	145.9727	3.6288e+04
	ADMM_{1e-6}	80	7.4000/16	90.8541	3.6288e+04
	ADMM_{1e-4}	75	2.5733/10	40.5114	3.6288e+04
	ADMM_{1e-2}	> 500	$\sim$	$\sim$	$\sim$
	InADMM	79	1.1899/2	31.5043	3.6288e+04
$(2.5*10^4, 5*10^4, 1\%)$	$\text{ADMM}_{Cholesky}$	83	$\sim$	4581.3164	4.6552e+03
	ADMM_{LSQR}	83	$\sim$	65.1848	4.6552e+03
	ADMM_{1e-10}	83	17.9277/25	121.7677	4.6552e+03
	ADMM_{1e-8}	83	12.6627/20	88.4123	4.6552e+03
	ADMM_{1e-6}	83	7.0723/14	53.7587	4.6552e+03
	ADMM_{1e-4}	81	2.6049/9	25.6452	4.6552e+03
	ADMM_{1e-2}	> 500	$\sim$	$\sim$	$\sim$
	InADMM	83	1.2892/2	22.1035	4.6552e+03
$(10^5, 1.5*10^5, 0.1\%)$	$\text{ADMM}_{Cholesky}$	$\sim$	$\sim$	> 5000	$\sim$
	ADMM_{LSQR}	70	$\sim$	115.3126	1.4368e+03
	ADMM_{1e-10}	70	22.0429/30	239.8781	1.4368e+03
	ADMM_{1e-8}	70	15.4714/24	171.0421	1.4368e+03
	ADMM_{1e-6}	71	8.4789/17	103.4054	1.4368e+03
	ADMM_{1e-4}	69	2.9565/10	47.2624	1.4368e+03
	ADMM_{1e-2}	> 500	$\sim$	$\sim$	$\sim$
	InADMM	71	1.3239/3	36.5243	1.4368e+03
$(10^5, 10^6, 0.01\%)$	$\text{ADMM}_{Cholesky}$	$\sim$	$\sim$	> 5000	$\sim$
	ADMM_{LSQR}	134	$\sim$	164.9873	289.2456
	ADMM_{1e-10}	134	10.179/14	197.8944	289.2456
	ADMM_{1e-8}	134	7.7239/11	151.0366	289.2456
	ADMM_{1e-6}	135	4.8370/9	104.9057	289.2456
	ADMM_{1e-4}	137	2.1679/5	71.1910	289.2456
	ADMM_{1e-2}	> 500	$\sim$	$\sim$	$\sim$
	InADMM	141	1.1418/2	59.3988	289.2430

and the proposed inexactness criterion automatically generates the good choice of accuracy of about 10^{-4} for solving the involved linear systems; and it automatically avoids too high or too low accuracy.

Table 4: Numerical Comparison on "news20" Dataset

Algorithm	Iteration	Mean CG Number	Max CG Number	Time	Obj
ADMM_{LSQR}	18	$\sim$	$\sim$	20.8350	7.3341e+03
ADMM_{1e-10}	19	18.1579	22	28.1866	7.3339e+03
ADMM_{1e-8}	19	13.9474	17	22.3447	7.3339e+03
ADMM_{1e-6}	19	19.1579	13	16.8641	7.3339e+03
ADMM_{1e-4}	18	5.7778	9	10.4031	7.3339e+03
ADMM_{1e-2}	110	0.7364	4	20.7784	7.3338e+03
InADMM	19	3.2632	4	8.1695	7.3340e+03

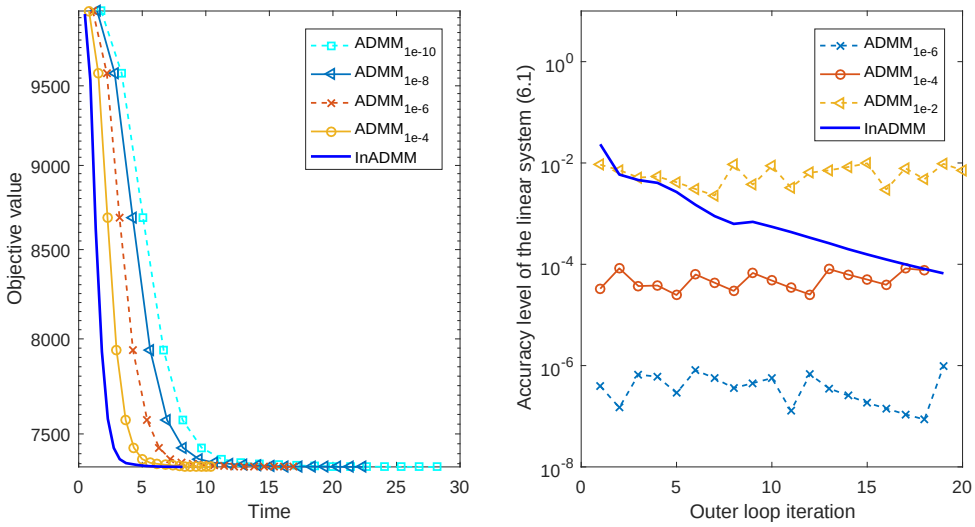

Figure 1: Objective value with respect to computing time and accuracy level of the linear system (6.1) with respect to the outer loop iteration on "news20" dataset.

6.2 Distributed LASSO

The generic LASSO model (1.2) also accounts for various distributed optimization models arising in multi-agent networks and hence it can be specified as concrete applications in this area. In a multi-agent network, the agents seek to collaborate to accomplish certain tasks. For example, distributed database servers may cooperate for parameter learning in order to fully exploit the data collected from individual servers; or a computation task may be executed by collaborative microprocessors with individual memories and storage spaces. We refer to [1, 4, 7, 8, 33, 44] for few references. In this subsection, we test some big datasets arising in such a distributed optimization problem and numerically show the efficiency of the proposed InADMM.

6.2.1 Model and specification of the application of InADMM

Some distributed optimization problem can be modelled as the LASSO model (1.2) with the specific sum form:

$$\min_x \sum_{i=1}^{N} \frac{1}{2} \|Q_i x - q_i\|_2^2 + \tau \|x\|_1, \tag{6.3}$$

where $x \in \mathbb{R}^n$ is the common decision variable, $\frac{1}{2}\|Q_i x - q_i\|_2^2$ is the cost function associated with agent i, Q_i is some data matrix (not necessarily to be full column rank) and $\|x\|_1$ reflects the sparsity character of x (see, e.g. [2]). Note that the penalty term $\|x\|_1$ can be replaced by more general ones such as the structured group sparsity regularization (see, e.g., [43]); but we do not discuss these more complicated cases in this paper. In the setting of distributed optimization, it is commonly assumed that each agent i only has knowledge about the local information Q_i and q_i. The challenge is to obtain, for each agent in the system, the optimal $x^\star$ of (6.3) using only local information and messages exchanged with neighbors [8, 33, 44].

Clearly, we can reformulate (6.3) as

$$\min_{x\in\mathbb{R}^n} \ \frac{1}{2}\sum_{i=1}^{N} \|Q_i x - q_i\|_2^2 + \tau\|x\|_1 \quad \Leftrightarrow \quad \begin{array}{ll} \min\limits_{\{x_i\in\mathbb{R}^n\}, x\in\mathbb{R}^n} & \frac{1}{2}\sum\limits_{i=1}^{N} \|Q_i x_i - q_i\|_2^2 + \tau\|x\|_1 \\ \text{s.t.} & x_i = x, \end{array} \tag{6.4}$$

so that we can apply various ADMM schemes. Note that the model (6.4) corresponds to the model (1.1) with the matrix A being a $(nN)\times(nN)$-dimensional identity matrix (full column rank) and $Q = \operatorname{diag}\{Q_1, \cdots, Q_N\}$.

For the distributed LASSO model (6.3), the x-subproblem (2.5a) can be specified as N smaller linear systems with each corresponding to an agent or server of a distributed network. That is, we can calculate x_i^{k+1} via the following process:

$$\begin{cases} \hat{H}_i \eta_i^{k+1} = \dfrac{1}{\beta} Q_i h_i^k, & \text{(6.5a)} \\ x_i^{k+1} = \dfrac{1}{\beta} Q_i \left(h_i^k - Q_i^T \eta_i^{k+1} \right), & \text{(6.5b)} \end{cases}$$

with

$$\hat{H}_i = \frac{1}{\beta} Q_i^T Q_i + I \quad \text{and} \quad h_i^k = Q_i^T q_i + \beta x^k + \lambda_i^k. \tag{6.6}$$

In (6.6), λ_i denotes the Lagrange multiplier associated with the constraint $x_i = x$. For big-data scenarios of the distributed LASSO model (6.3), the individual matrix Q_i may be still of huge dimension though it may have some special structures such as the sparsity. Therefore, as the inexactness criterion (2.9), we should consider solving the linear systems (6.5a) inexactly for $i = 1, \cdots, N$. Let us define the residual of the linear system (6.5a) as

$$e_k^i(\eta_i) := \frac{1}{\beta} Q_i h_i^k - \hat{H}_i \eta_i, \quad i = 1, \cdots, N.$$

Moreover, we choose σ to satisfy

$$0 < \sigma < \frac{1}{1 + \frac{\max\{\|Q_i\|_2\}}{\sqrt{2\beta}}} = \frac{1}{1 + \frac{\|Q\|_2}{\sqrt{2\beta}}} \in (0, 1), \tag{6.7}$$

where the equation holds because $Q = \text{diag}\{Q_1, \cdots, Q_N\}$. Then, we suggest solving the linear system (6.5a) inexactly subject to the inexactness criterion

$$\left\| e_k^i \left(\eta_i^{k+1} \right) \right\|_2 \leq \sigma \cdot \left\| e_k^i \left(\eta_i^k \right) \right\|_2, \quad i = 1, \cdots, N.$$

Note that if we take all the distributed matrices Q_i and vectors q_i into consideration for $i = 1, 2, \cdots, N$, it holds that

$$\begin{aligned} \left\| e_k \left(\eta^{k+1} \right) \right\|_2 &= \left\| \hat{H}\eta^{k+1} - \frac{1}{\beta} Q h^k \right\|_2 = \sqrt{\sum_{i=1}^{N} \left\| \hat{H}_i \eta_i^{k+1} - \frac{1}{\beta} Q_i h_i^k \right\|_2^2} \leq \sqrt{\sigma^2 \sum_{i=1}^{m} \left\| \hat{H}_i \eta_i^k - \frac{1}{\beta} Q_i h_i^k \right\|_2^2} \\ &\leq \sigma \cdot \left\| \hat{H}\eta^k - \frac{1}{\beta} Q h^k \right\|_2 = \sigma \cdot \left\| e_k \left(\eta^k \right) \right\|_2. \end{aligned} \tag{6.8}$$

Hence, (6.8) is a specification of the one (2.9) when the general LASSO model (1.2) is specified as the distributed LASSO model (6.3).

To show the necessity and efficiency of the automatically adjustable inexactness criterion (6.8), as in Section 6.1, we also compare it with the case where the linear systems (6.5a) is solved either exactly or up to a-prior fixed accuracy levels. More specifically, we compare ADMM$_{LSQR}$ which means the linear system (6.5a) is solved exactly by the LSQR; and ADMM$_{1e-t}$ which means the linear systems (6.5a) is solved subject to the inexactness criterion with a fixed accuracy level:

$$\frac{\left\| \hat{H}_i \eta_i - \frac{1}{\beta} Q_i h_i^k \right\|_2}{\left\| \frac{1}{\beta} Q_i h_i^k \right\|_2} \leq 10^{-t}.$$

We test the cases where $t = 2, 4, 6, 8, 10$. Also, the InADMM and ADMM$_{1e-t}$ are terminated only when the generated objective function values are better than those found by ADMM$_{LSQR}$. Note that the extremely high dimensionality of this dataset prevents us to execute the Cholesky decomposition and thus we do not compare the case where the linear system (6.5a) is solved directly by the Cholesky decomposition.

6.2.2 Numerical results

We test the real big dataset: "url" dataset[5] in [30], and "avazu-app"[6]dataset in [26]. For both of the datasets, their dimensions are much higher than those of the "RCV1" and "news20" datasets.

The "url" dataset contains 121 days directory for malicious URLs (spam, phishing, exploits, and so on) detection and the total dataset size is about 470MB. The data sample has 3,231,961 features, and $2,396,130$ data samples for the total 121 days. In this experiment, because of our limited computation infrastructure, we only consider the cases of the first 10/15/20 days from the whole data, and treat each day's dataset as a subsystem. As a result, the dataset dimension of each subsystem is about $15,000 \times 2,396,130$.

For the "url" dataset, the values of τ, β and τ are calculated by the formulas $\tau = \max\left\{0.1 \cdot \|Q_i^T q_i\|_\infty\right\}_{i=1}^N$, $\beta = 0.05 \max\left\{\|Q_i^T q_i\|_\infty\right\}_{i=1}^N$ and $\sigma = \min\left\{\sigma_i = \frac{0.99}{1+\frac{\|Q_i\|_2}{\sqrt{2\beta}}}\right\}$, respectively. We consider the distributed scenarios of $N = 10, 15, 20$; and correspondingly the values of these constants are listed in Table 5.

[5] http://www.sysnet.ucsd.edu/projects/url/

[6] http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html♯avazu

Table 5: Values of τ, β and σ for "url" dataset

N	τ	β	σ
N=10	766.6	383.3	0.0198
N=15	775.6	387.8	0.0198
N=20	775.6	387.8	0.0198

Some results for the "url" dataset are reported in Table 6. As observed previously, a too-high accuracy level for the internally nested iterations such as ADMM_{1e-10} and ADMM_{1e-8} slows down the convergence; while a too-low accuracy level such as ADMM_{1e-2} and ADMM_{1e-4} does not guarantee the convergence. For this dataset, it turns out that 10^{-6} is appropriate for the internally nested iterations; and if this accuracy happens to be found, the resulting numerical performance is very competitive. But again there is no clue at all to discern this appropriate level of accuracy in advance. On the contrary, the proposed inexactness criterion (6.8) can automatically find this appropriate level of accuracy. To see the efficiency of InADMM more clearly, in Figure 2 we plot the evolution of the objective function values with respect to the computing time, and the evolution of the mean of inner-loop iteration numbers with respect to the outer-loop iterations. We see that generally the CG steps for the internally nested iterations to ensure the convergence of the InADMM are quite stable. In Figure 3, we also plot the evolutions of the primal and dual residuals of the model (6.4) with respect to the outer-loop iterations.

The "avazu-app" dataset is used in a competition on click-through rate prediction jointly hosted by Avazu and Kaggle in 2014. The participants were asked to learn a model from the first 10 days of advertising log, and predict the click probability for the impressions on the 11th day. This dataset contains 1,000,000 features and 14,596,137 samples; the total dataset size is about 394MB. We split this dataset into 29 groups (hence, $N = 29$ in (6.3)); the first 28 groups have $500,000$ samples and the last one has $596,137$ samples. Accordingly, the dataset dimension of each subsystem is $500,000 \times 1,000,000$ for the first 28 groups and $596,137 \times 1,000,000$ for the last one. Values of the parameters τ, β and σ are computed by the same formulas as those for the "url" dataset; and they are $\tau = 2219.0, \beta = 1109.5, \sigma = 0.0825$, respectively. Other implementation details are the same as those mentioned in Section 6.1.

Some numerical results for testing the "avazu-app" dataset are reported in Table 7. Similar conclusions as those for the previous experiments can be derived. In particular, for this dateset, it seems that 10^{-4}, instead of 10^{-6} for the "url" dataset, is appropriate for the internally nested iterations. Because of the extremely high dimensionality of the variables, slightly increasing the accuracy for the internally nested iterations results in significantly additional computation and thus slows down the overall speed very much. For such a big dataset, it is more evident to use the proposed inexactness criterion (6.8) when implementing the ADMM, rather than a trail-and-error procedure of seeking an appropriate level of accuracy.

7 Conclusions

In this paper, we discuss how to implement the alternating direction method of multipliers (ADMM) to big datasets in the convex programming context, with an emphasis on the problem of least absolute shrinkage and selection operator (LASSO). We show that the system of linear equations arising at each iteration of the ADMM should be solved inexactly and an automatically adjustable

Table 6: Numerical Comparison on "url" Dataset

N	Algorithm	Iteration	Mean CG Number	Max CG Number	Time	Obj
$N = 10$	ADMM_{LSQR}	31	$\sim$	$\sim$	774.7665	5.9316e+3
	ADMM_{1e-10}	31	35.8226	52	594.7653	5.9316e+3
	ADMM_{1e-8}	32	22.5344	37	398.8079	5.9318e+3
	ADMM_{1e-6}	30	10.3400	23	200.5246	5.9138e+3
	ADMM_{1e-4}	> 500	$\sim$	$\sim$	$\sim$	$\sim$
	ADMM_{1e-2}	> 500	$\sim$	$\sim$	$\sim$	$\sim$
	InADMM	30	7.4267	14	165.1480	5.9268e+3
$N = 15$	ADMM_{LSQR}	40	$\sim$	$\sim$	1430.7386	7.8744e+3
	ADMM_{1e-10}	40	33.8533	57	1087.9498	7.8744e+3
	ADMM_{1e-8}	40	21.1567	40	716.4767	7.8744e+3
	ADMM_{1e-6}	39	9.2137	23	361.1047	7.8730e+3
	ADMM_{1e-4}	> 500	$\sim$	$\sim$	$\sim$	$\sim$
	ADMM_{1e-2}	> 500	$\sim$	$\sim$	$\sim$	$\sim$
	InADMM	36	7.9537	17	313.6074	7.8723e+3
$N = 20$	ADMM_{LSQR}	34	$\sim$	$\sim$	1655.6177	9.5673e+3
	ADMM_{1e-10}	34	34.4029	57	1258.1451	9.5673e+3
	ADMM_{1e-8}	34	21.5441	40	889.1597	9.5673e+3
	ADMM_{1e-6}	32	9.8594	23	422.1671	9.5671e+3
	ADMM_{1e-4}	> 500	$\sim$	$\sim$	$\sim$	$\sim$
	ADMM_{1e-2}	> 500	$\sim$	$\sim$	$\sim$	$\sim$
	InADMM	37	8.0527	24	441.9357	9.5650e+3

Table 7: Numerical Comparison on "avazu-app" Dataset

Algorithm	Iteration	Mean CG Number	Max CG Number	Time	Obj
ADMM_{LSQR}	36	$\sim$	$\sim$	1446.5865	7.2533e+05
ADMM_{1e-10}	37	18.6048	26	2887.6285	7.2533e+05
ADMM_{1e-8}	37	13.4921	21	2178.0520	7.2533e+05
ADMM_{1e-6}	36	8.2299	16	1425.0370	7.2533e+05
ADMM_{1e-4}	30	3.4747	12	633.4167	7.2533e+05
ADMM_{1e-2}	> 500	$\sim$	$\sim$	$\sim$	$\sim$
InADMM	38	3.1343	8	809.2827	7.2533e+05

inexactness criterion is proposed. This inexactness criterion automatically avoids too high or too low accuracy for the subproblems. We also specify the safe-guard iteration numbers for several standard numerical linear algebra solvers when they are used for the subproblems; and thus make the inexact implementation of ADMM with an internally nested iterative procedure fully automatic. Existing convergence results for the exact version of ADMM are not applicable. Hence, the convergence is proved and the worst-case convergence rate measured by the iteration complexity is established for the proposed inexact version of ADMM with an internally nested iterative procedure. Some real big datasets with millions of variables are tested to numerically show the efficiency of the inexact version

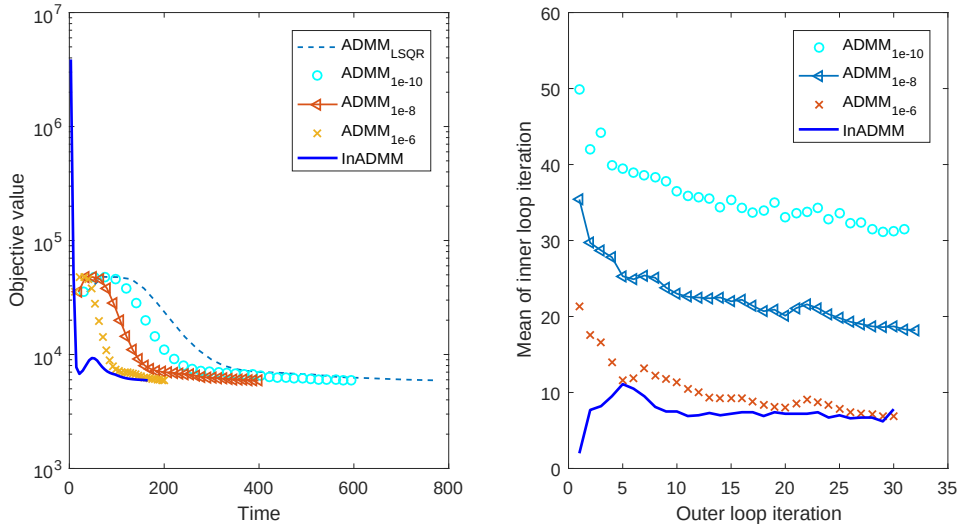

Figure 2: Objective value with respect to the computing time and mean of inner loop iteration number with respect to the outer loop iteration number on "url" dataset ($N = 10$).

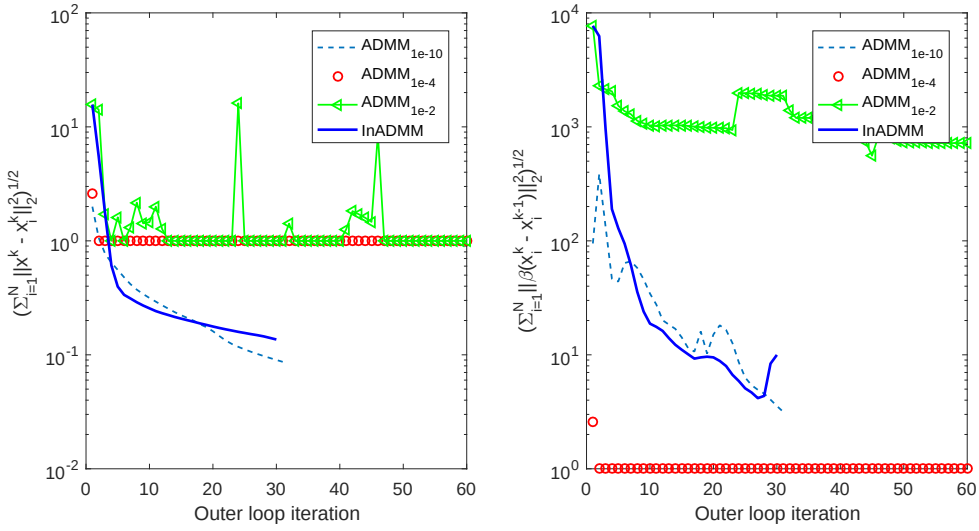

Figure 3: Norms of primal residual and dual residual with respect to the outer loop iteration number on "url" dataset ($N = 10$).

of ADMM. These results show that usually only a few steps for the internally nested iterations are sufficient to guarantee the convergence of the inexact version of ADMM, despite of the high dimensionality of their variables. Hence, the inexact implementation of ADMM to big datasets can be significantly accelerated if the subproblems are solved inexactly subject to an appropriate inexactness criterion; meanwhile the convergence can be rigorously guaranteed.

References

[1] G. R. Andrews. *Foundations of Multithreaded, Parallel, and Distributed Programming*. Addison-Wesley, 2007.

[2] F. Bach, R. Jenatton, J. Mairal, and G. Obozinski. Optimization with sparsity-inducing penalties. *Foundations and Trends in Machine Learning*, 4(1):1–106, 2011.

[3] R. Baraniuk and P. Steeghs. Compressive radar imaging. In *2007 IEEE Radar Conference*, pages 128–133, 2007.

[4] R. Bekkerman, M. Bilenko, and J. Langford. *Scaling up Machine Learning: Parallel and Distributed Approaches*. Cambridge University Press, 2011.

[5] 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):1–122, 2010.

[6] T. Chan and R. Glowinski. *Finite element approximation and iterative solution of a class of mildly non-linear elliptic equations*. Computer Science Department, Stanford University, 1978.

[7] T.-H. Chang, M. Y. Hong, and X. F. Wang. Multi-agent distributed optimization via inexact consensus ADMM. *IEEE Transactions on Signal Processing*, 63(2):482–497, 2014.

[8] J. Chen and A. H. Sayed. Diffusion adaptation strategies for distributed optimization and learning over networks. *IEEE Transactions on Signal Processing*, 60(8):4289–4305, 2012.

[9] M. Heredia Conde. Fundamentals of compressive sensing. In *Compressive Sensing for the Photonic Mixer Device: Fundamentals, Methods and Results*, pages 89–205. Springer Fachmedien Wiesbaden, 2017.

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

[11] J. Eckstein and W. Yao. Understanding the convergence of the alternating direction method of multipliers: Theoretical and computational perspectives. *Pacific Journal of Optimization*, 11:619–644, 2015.

[12] J. Eckstein and W. Yao. Relative-error approximate versions of douglas–rachford splitting and special cases of the admm. *Mathematical Programming*, to appear.

[13] F. Facchinei and J.-S. Pang. *Finite-dimensional variational inequalities and complementarity problems*. Springer Science & Business Media, 2007.

[14] M. Fortin and R. Glowinski. Augmented lagrangian methods in quadratic programming. *Studies in Mathematics and Its Applications*, 15:1–46, 1983.

[15] J. Friedman, T. Hastie, and R. Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.

[16] D. Gabay. Applications of the method of multipliers to variational inequalities. *Studies in Mathematics and its Applications*, 15:299–331, 1983.

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

[18] R. Glowinski. On alternating direction methods of multipliers: A historical perspective. *Modeling, Simulation and Optimization for Science and Technology*, W. Fitzgibbon and Y. A. Kuznetsov and P. Neittaanm ki and O. Pironneau(Springer Netherlands):59–82, 2014.

[19] R. Glowinski and A. Marrocco. Approximation par éléments finis d'ordre un et résolution par pénalisation-dualité d'une classe de problèmes non linéaires. *R.A.I.R.O., R2*, 60(8):41–76, 1975.

[20] R. Glowinski and P. Le Tallec. *Augmented Lagrangian and Operator Splitting Methods in Nonlinear Mechanics*. Studies in Applied and Numerical Mathematics, 1989.

[21] T. Goldstein and S. Osher. The split Bregman method for ℓ_1-regularized problems. *SIAM Journal on Imaging Sciences*, 2(2):323–343, 2009.

[22] B. S. He, L.-Z. Liao, D. R. Han, and H. Yang. A new inexact alternating directions method for monotone variational inequalities. *Mathematical Programming*, 92(1):103–118, 2002.

[23] B. S. He and H. Yang. Some convergence properties of a method of multipliers for linearly constrained monotone variational inequalities. *Operations Research Letters*, 23(3):151–161, 1998.

[24] B. S. He and X. M. Yuan. On the $O(1/n)$ convergence rate of the Douglas-Rachford alternating direction method. *SIAM Journal on Numerical Analysis*, 50(2):700–709, 2012.

[25] B. S. He and X. M. Yuan. On non-ergodic convergence rate of Douglas-Rachford alternating direction method of multipliers. *Numerische Mathematik*, 130(3):567–577, 2015.

[26] Y. C. Juan, Y. Zhuang, W.-S. Chin, and C.-J. Lin. Field-aware factorization machines for CTR prediction. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pages 43–50. ACM, 2016.

[27] S. S. Keerthi and D. DeCoste. A modified finite newton method for fast solution of large scale linear SVMs. *Journal of Machine Learning Research*, 6(3):341–361, 2005.

[28] D. D. Lewis, Y. M. Yang, T. G. Rose, and F. Li. RCV1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5(4):361–397, 2004.

[29] M. Lustig, D. L. Donoho, and J. M. Pauly. Sparse MRI: The application of compressed sensing for rapid MR imaging. *Magnetic Resonance in Medicine*, 58(6):1182–1195, 2007.

[30] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker. Identifying suspicious URLs: an application of large-scale online learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 681–688. ACM, 2009.

[31] R. Mises and H. Pollaczek-Geiringer. Praktische verfahren der gleichungsauflsung . *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift fr Angewandte Mathematik und Mechanik*, 9(2):152–164, 1929.

[32] R. Monteiro and B. Svaiter. Iteration-complexity of block-decomposition algorithms and the alternating direction method of multipliers. *SIAM Journal on Optimization*, 23(1):475–507, 2013.

[33] A. Nedic, A. Ozdaglar, and P. A. Parrilo. Constrained consensus and optimization in multi-agent networks. *IEEE Transactions on Automatic Control*, 55(4):922–938, 2010.

[34] J. Neumann, C. Schnörr, and G. Steidl. Combined SVM-based feature selection and classification. *Machine Learning*, 61(1):129–150, 2005.

[35] M. K. Ng, F. Wang, and X. M. Yuan. Inexact alternating direction methods for image recovery. *SIAM Journal on Scientific Computing*, 33(4):1643–1668, 2011.

[36] C. Paige and M. Saunders. LSQR: An algorithm for sparse linear equations and sparse least squares. *ACM Transactions on Mathematical Software*, 8(1):43–71, 1982.

[37] Benjamin Recht, Maryam Fazel, and Pablo A Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM review*, 52(3):471–501, 2010.

[38] R. T. Rockafellar and R. J-B Wets. *Variational Analysis*, volume 317. Springer Science & Business Media, 2009.

[39] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, pages 267–288, 1996.

[40] L. N. Trefethen and D. Bau III. *Numerical Linear Algebra*, volume 50. SIAM, 1997.

[41] R. S. Varga. *Matrix Iterative Analysis*, volume 27. Springer Science & Business Media, 2009.

[42] M. A. Woodbury. *Inverting modified matrices*. Princeton University, Princeton, New Jersey, 1950.

[43] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.

[44] M. Zhu and S. Martinez. On distributed convex optimization under inequality and equality constraints. *IEEE Transactions on Automatic Control*, 57(1):151–164, 2012.