SPARSE APPROXIMATIONS WITH INTERIOR POINT METHODS*

VALENTINA DE SIMONE†, DANIELA DI SERAFINO‡, JACEK GONDZIO§, SPYRIDON POUGKAKIOTIS§, AND MARCO VIOLA†

VERSION 3 – November 24, 2021

Abstract. Large-scale optimization problems that seek sparse solutions have become ubiquitous. They are routinely solved with various specialized first-order methods. Although such methods are often fast, they usually struggle with not-so-well conditioned problems. In this paper, specialized variants of an interior point-proximal method of multipliers are proposed and analyzed for problems of this class. Computational experience on a variety of problems, namely, multi-period portfolio optimization, classification of data coming from functional Magnetic Resonance Imaging, restoration of images corrupted by Poisson noise, and classification via regularized logistic regression, provides substantial evidence that interior point methods, equipped with suitable linear algebra, can offer a noticeable advantage over first-order approaches.

Key words. Sparse Approximations, Interior Point Methods, Proximal Methods of Multipliers, Nonlinear Convex Programming, Solution of KKT Systems, Portfolio Optimization, Image Restoration, Classification in Machine Learning.

AMS subject classifications. 65K05, 90C51, 90C25, 65F10, 65F08, 90C90.

1. Introduction.

We are concerned with the efficient solution of a class of problems which are very large and are expected to yield sparse solutions. In practice, the sparsity is often induced by the presence of ℓ_1 norm terms in the objective. We assume that a general problem of the following form

$$\begin{aligned} \min_{x} \quad & f(x) + \tau_1 \|x\|_1 + \tau_2 \|Lx\|_1 \\ \text{s.t.} \quad & Ax = b, \end{aligned} \tag{1.1}$$

needs to be solved, where $f : \mathbb{R}^n \mapsto \mathbb{R}$ is a twice continuously differentiable convex function, $L \in \mathbb{R}^{l \times n}$, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $m \leq n$, and $\tau_1, \tau_2 > 0$. We are particularly interested in problems for which $f(x)$ displays some level of separability. The terms $\|x\|_1$ and $\|Lx\|_1$ induce sparsity in the vector x and/or in some (possibly redundant) dictionary Lx. Numerous real-life problems can be recast into the form (1.1). Among the various application areas, one can find portfolio optimization [55], signal and image processing [18, 67], classification in statistics [73] and machine learning [76], inverse problems [74] and compressed sensing [15], to mention just a few.

Optimization problems arising in these applications are usually solved by different specialized variants of first-order methods. Indeed, highly specialized and tuned to a

*Funding: this work was funded by various institutes and research programs. V. De Simone, D. di Serafino and M. Viola were supported by the Istituto Nazionale di Alta Matematica, Gruppo Nazionale per il Calcolo Scientifico (INdAM-GNCS), and by the V:ALERE Program of the University of Campania "L. Vanvitelli", Italy. S. Pougkakiotis was supported by a Principal's Career Development scholarship from the University of Edinburgh, as well as a scholarship from A. G. Leventis Foundation. J. Gondzio and S. Pougkakiotis were also supported by the Google project "Fast $(1 + x)$-order Methods for Linear Programming". We wish to remark that this study does not have any conflict of interest to disclose.

†Department of Mathematics and Physics, University of Campania "L. Vanvitelli", Caserta, Italy (valentina.desimone@unicampania.it, marco.viola@unicampania.it).

‡Department of Mathematics and Applications, University of Naples Federico II, Napoli, Italy (daniela.diserafino@unina.it).

§School of Mathematics and Maxwell Institute for Mathematical Sciences, University of Edinburgh, Edinburgh, United Kingdom (J.Gondzio@ed.ac.uk, S.Pougkakiotis@sms.ed.ac.uk).

narrow class of problems, first-order methods often outperform standard off-the-shelf second-order techniques; the latter might be too expensive or might struggle with excessive memory requirements. Such comparisons are not fair though. With this paper we hope to change the incorrect opinion on the second-order methods.

Various second-order approaches have been proposed in the literature for problems of the form of (1.1). In particular, one might employ proximal (projected) Newton-type methods (see [50, 71]) in which a proximal term is added to deal with the non-smooth part of the objective function (unless its minimum-norm subgradient has a closed form solution, in which case the proximal term can be excluded). Alternatively, such problems can be solved by means of standard semi-smooth Newton methods (see [47, 48] and the references therein) or semi-smooth Newton methods combined with the augmented Lagrangian method (see, e.g., [52]). The aforementioned approaches employ line-search schemes that allow one to show linear or local superlinear convergence, given certain assumptions. For methods involving proximal terms, superlinear convergence is only guaranteed when the associated penalty parameters increase to infinity.

Here we consider Interior Point Methods (IPMs), which exhibit better convergence (in practice and in theory), at the expense of worse conditioning of the associated linear systems that have to be solved at every IPM iteration. When efficiently implemented and specialized to a particular application, interior point methods offer an attractive alternative. They can be equally (or more) efficient than the best first-order methods available, and they deliver unmatched robustness and reliability.

The specializations of interior point methods proposed in this paper do not go beyond what has been commonly exploited by first-order methods. Namely we propose:

- to exploit special features of the problems in the linear algebra of IPMs, and
- to take advantage of the expected sparsity of the optimal solution.

In order to achieve our goals we propose to convert sparse approximation problems to standard smooth nonlinear convex programming ones by replacing the ℓ_1 norm terms with a usual modeling trick in which an absolute value $|a|$ is substituted with the sum of two non-negative parts, $|a| = a^+ + a^-$, where $a^+ = \max\{a, 0\}$ and $a^- = \max\{-a, 0\}$. By introducing the auxiliary variable $d = Lx \in \mathbb{R}^l$, problem (1.1) is then transformed into the following one:

$$
\begin{aligned}
\min_{x^+,x^-,d^+,d^-} \quad & f(x^+ - x^-) + \tau_1(e_n^\top x^+ + e_n^\top x^-) + \tau_2(e_l^\top d^+ + e_l^\top d^-), \\
\text{s.t.} \quad & A(x^+ - x^-) = b, \\
& L(x^+ - x^-) = d^+ - d^-, \\
& x^+, x^-, d^+, d^- \geq 0,
\end{aligned}
\tag{1.2}
$$

where $x^+, x^- \in \mathbb{R}^n$ are such that $x = x^+ - x^-$, $d^+, d^- \in \mathbb{R}^l$ are such that $d = d^+ - d^-$, and $e_n \in \mathbb{R}^n$, $e_l \in \mathbb{R}^l$ are vectors with all entries equal to 1. It is worth observing that (1.1) and (1.2) are equivalent; indeed the presence of linear terms which penalize for the sum of positive and negative parts of vectors x and d guarantees that at optimality only one of the split variables can take a nonzero value. We also note that the number of variables is greater than or equal to the number of equality constraints in (1.2). Although (1.2) is larger than (1.1) because the variables have been replicated and new constraints have been added, it is in a form eligible to a straightforward application of an interior point method. We expect that the well-known ability of IPMs to handle large sets of linear equality and non-negativity constraints will compensate for this increase of the problem dimension.

IPMs employ Newton method to solve a sequence of so-called logarithmic barrier subproblems (see Section 2 for details). In standard implementations of IPMs this requires many involved linear algebra operations (building and inverting the Hessian matrix), and for large problems it might become prohibitively expensive. In this paper we demonstrate that the use of inexact Newton method [8, 14, 38] combined with a knowledgeable choice and appropriate tuning of linear algebra solvers (see [27, 30, 37, 40] and the references therein) is the key to success when developing an IPM specialized to a particular class of problems. We also demonstrate an attractive ability of IPMs to select important variables and prematurely drop the irrelevant ones, a feature which is very well suited to solving sparse approximation problems in which the majority of variables are expected to be zero at optimality. It is worth mentioning at this point that our understanding of the features of IPMs applied to sparse approximation problems benefitted from the earlier studies which focused on compressed sensing problems [33, 34].

Ultimately, we provide computational evidence that IPMs can be more efficient than methods which are routinely used for the solution of sparse approximation problems by exploiting only first-order information.

Notation. Throughout this paper we use lowercase Roman and Greek fonts to indicate scalars and vectors (the nature is clear from the context). Capital italicized Roman fonts are used to indicate matrices. Superscripts are used to denote the components of a vector/matrix. As an example, given $M \in \mathbb{R}^{m\times n}$, $v \in \mathbb{R}^n$, $\mathcal{R} \subseteq \{1,\dots,m\}$, and $\mathcal{C} \subseteq \{1,\dots,n\}$, we set $v^{\mathcal{C}} := (v^i)_{i\in\mathcal{C}}$ and $M^{\mathcal{R},\mathcal{C}} := \left(m^{ij}\right)_{i\in\mathcal{R},j\in\mathcal{C}}$, where v^i is the i-th entry of v and m^{ij} the (i,j)-th entry of M. We use $\lambda_{\min}(B)$ ($\lambda_{\max}(B)$, respectively) to denote the minimum (maximum) eigenvalue of an arbitrary square matrix B with real eigenvalues. Similarly, $\sigma_{\min}(B)$ ($\sigma_{\max}(B)$, respectively) denotes the minimum (maximum) singular value of an arbitrary rectangular matrix B. We use $B \succ 0$ to indicate that a square matrix B is symmetric positive definite. We use e_n and 0_n to denote a column vector of size n with all entries equal to 1 and 0, respectively. Moreover, we use I_n to indicate the identity matrix of size n and $0_{m,n}$ to denote the zero matrix of size $m \times n$. We use subscripts to denote the elements of a sequence, e.g., $\{x_k\}$. Norms $\|\cdot\|$ are ℓ_2. Other norms are identified by adding suitable subscripts. For any finite set $\mathcal{A}$, we denote by $|\mathcal{A}|$ its cardinality. Finally, when referring to convex programming problems, we implicitly assume that the problems are linearly constrained.

Structure of the article. The rest of this article is organized as follows. In Section 2 we briefly describe IPMs for convex programming, focusing in particular on the *Interior Point-Proximal Method of Multipliers* (IP-PMM), which is used in the subsequent sections. The choice of IP-PMM is motivated by the fact that it merges an infeasible IPM with the Proximal Method of Multipliers (PMM), in order to keep the fast and reliable convergence properties of IPMs and the strong convexity of the PMM subproblems, thus achieving better efficiency and robustness than both methods. In this section we also outline the testing environment used throughout the paper. In Sections 3 to 6 we present four applications formulated as optimization problems with sparsity sought in the solutions, and recast them in the form (1.2). In detail, in Section 3 we focus on a multi-period portfolio selection strategy, in Section 4 on the classification of data coming from functional Magnetic Resonance Imaging (fMRI), in Section 5 on the restoration of images corrupted by Poisson noise, and in Section 6 on linear classification through regularized logistic regression. The first two applications

yield convex quadratic programming problems, while the remaining ones yield general nonlinear convex programming problems. For each application, we provide a brief description of its mathematical model and explain how IP-PMM is specialized for that case in terms of linear algebra solvers, including variable dropping strategies to help sparsification; we also show the results of computational experiments, including comparisons with state-of-the art methods widely used by the scientific community on the selected problems.

2. Interior Point Methods for Convex Programming. We consider the following convex programming problem:

$$\min_x \ f(x), \quad \text{s.t.} \ \ Ax = b, \ \ x \geq 0, \tag{2.1}$$

where $x \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$, and $f \colon \mathbb{R}^n \mapsto \mathbb{R}$ is a twice differentiable convex function. Using the Lagrangian duality theory [11], and defining a function $F(w) : \mathbb{R}^{2n+m} \mapsto \mathbb{R}^{2n+m}$, we write the KKT (optimality) conditions as follows:

$$F(w) = \begin{bmatrix} \nabla f(x) - A^\top y - z \\ Ax - b \\ XZe_n \end{bmatrix} = \begin{bmatrix} 0_n \\ 0_m \\ 0_n \end{bmatrix}, \tag{2.2}$$

where $y \in \mathbb{R}^m$ and $z \in \mathbb{R}^n$ are the Lagrange multipliers corresponding to the equality and inequality constraints respectively, while X, $Z \in \mathbb{R}^{n \times n}$ denote the diagonal matrices with diagonal entries x^i and z^i (respectively), $\forall\, i \in \{1, \ldots, n\}$.

Problem (2.1) can be solved using a primal-dual IPM. There are numerous variants of IPMs and the reader is referred to [37] for an extended literature review. IPMs handle the non-negativity constraints of the problems with logarithmic barriers in the objective. That is, at each iteration k, we choose a *barrier parameter* μ_k and form the *logarithmic barrier problem*:

$$\min_x \ f(x) - \mu_k \sum_{j=1}^{n} \ln x^j, \quad \text{s.t.} \ \ Ax = b. \tag{2.3}$$

Then, a damped Newton method (or possibly an inexact variant of it [8, 38, 60]) is usually employed in order to approximately solve problem (2.3). Applying it to the optimality conditions of (2.3), and further forming the augmented system (as is done in Section 2.2), we obtain a system of the following form:

$$\begin{bmatrix} -(\nabla^2 f(x_k) + \Theta_k^{-1}) & A^\top \\ A & 0_{m,m} \end{bmatrix} \begin{bmatrix} \Delta x_k \\ \Delta y_k \end{bmatrix} = \begin{bmatrix} \nabla f(x_k) - A^\top y_k - \sigma_k \mu_k X_k^{-1} e \\ b - Ax_k \end{bmatrix}, \tag{2.4}$$

where $\Theta_k = X_k Z_k^{-1}$ and the entries of x_k and z_k are maintained positive throughout the algorithm, allowing the use of the logarithmic barrier (see Section 2.2 for details). One can observe that the matrix Θ_k contains some very large and some very small elements close to optimality. Hence, the matrix in (2.4) becomes increasingly ill-conditioned, as the method progresses. Notice that as $\mu_k \to 0$, an optimal solution of (2.3) converges to an optimal solution of (2.1). Polynomial convergence of such methods (with respect to the number of variables n), for various classes of problems, has been proved multiple times in the literature (see for example [60, 84]).

A system like (2.4) can either be solved directly (using an appropriate factorization, as in [1, 35, 64]) or iteratively (using an appropriate Krylov subspace method, as

in [9, 13, 27, 40]). While the former approach is very general, it becomes problematic as the problem size increases. On the other hand, iterative methods (accompanied by appropriate preconditioners) may be difficult to generalize. However, if applied to specific classes of problems, they make possible solving huge-scale instances, by avoiding the explicit storage of the problem matrices.

2.1. Regularization in IPMs. In the context of IPMs, it is often beneficial to include some regularization, in order to improve the spectral properties of the system matrix in (2.4). For example, notice that if the constraint matrix A is rank deficient, then the matrix in (2.4) might not be invertible. The latter can be immediately addressed by the introduction of a dual regularization, say $\delta > 0$, ensuring that $\text{rank}([A\ \delta I_m]) = m$. The introduction of a primal regularization, say $\rho > 0$, can ensure that the matrix $\nabla^2 f(x_k)+\Theta_k^{-1}+\rho I_n$, has eigenvalues that are bounded away from zero, and hence a significantly better worst-case conditioning than that of $\nabla^2 f(x_k) + \Theta_k^{-1}$. To produce a diagonal term in the (2,2) block Vanderbei added artificial variables to all the constraints [75]. Saunders and Tomlin [69, 70] achieved a similar result for the (1,1) and (2,2) blocks, by adding Tikhonov-type regularization terms to the original problem. In the aforementioned approaches, in order to guarantee that the solution of the original problem is retrieved, one has to ensure that the regularization parameters are smaller than some unknown (nonzero) value, in which case the regularization can be shown to be *exact* (see [36], and the references therein).

In later works, these Tikhonov-type regularization methods were replaced by algorithmic regularization schemes. In particular, one can observe that a very natural way of introducing primal regularization to problem (2.1), is through the application of the primal proximal point method. Similarly, dual regularization can be incorporated through the application of the dual proximal point method. This is a well-known fact. The authors in [1] presented a primal-dual regularized IPM for convex Quadratic Programming (QP), and interpreted this regularization as the application of the proximal point method. Subsequently, the authors in [35] developed a primal-dual regularized IPM, which applies PMM to solve convex QP problems, and employs a single IPM iteration for approximating the solution of each PMM subproblem. There, global convergence of the method was proved, under some assumptions. A variation of the method proposed in [35] is given in [63], where general non-diagonal regularization matrices are employed, as a means of further improving factorization efficiency. Then, the authors in [64] proposed an IP-PMM and proved (under standard assumptions) that it achieves convergence to an ϵ-optimal solution in a polynomial (in n) number of iterations for convex QP problems and for linear semidefinite programming problems (see [65]), including cases when the associated Newton systems are solved iteratively. Finally, a similar algorithm was proposed for general nonlinear programming problems in [3], and was shown to be convergent under standard assumptions. In all these cases, algorithmic regularization ensures stability, while allowing one to retrieve the solution of the original problem.

2.2. Interior Point-Proximal Method of Multipliers. In this subsection, we derive an IP-PMM suitable for solving convex programming problems. The method is based on the developments in [64]. We consider the following primal problem (which can be equivalently formulated as (2.1), by adding some additional constraints):

$$\min_x \ f(x), \quad \text{s.t.} \ \ Ax = b, \ \ x^{\mathcal{I}} \geq 0, \ \ x^{\mathcal{F}} \text{ free}, \tag{2.5}$$

where $\mathcal{I} \subseteq \{1, \ldots, n\}$, $\mathcal{F} = \{1, \ldots, n\} \setminus \mathcal{I}$. In the above problem, we assume that the dimensions of the involved matrix are the same as those in (2.1). Effectively, an

IP-PMM arises by merging PMM with an infeasible IPM. For that purpose, assume that, at some iteration k of the method, we have available an estimate η_k for an optimal Lagrange multiplier vector y^* associated to the equality constraints of (2.5). Similarly, we denote by ζ_k an estimate of a primal solution x^*. Now, we define the proximal penalty function that has to be minimized at the k-th iteration of the PMM, for solving (2.5), given the estimates $\eta_k,\ \zeta_k$:

$$\mathcal{L}^{PMM}_{\rho_k,\delta_k}(x;\zeta_k,\eta_k) = f(x) - \eta_k^\top(Ax-b) + \frac{1}{2\delta_k}\|Ax-b\|_2^2 + \frac{\rho_k}{2}\|x-\zeta_k\|_2^2,$$

with $\{\delta_k\},\ \{\rho_k\}$ two positive non-increasing sequences of penalty parameters. Following [64], we require that these parameters decrease at the same rate as μ_k; however, in practice we never allow these values to be reduced below a certain appropriately chosen threshold. For more details on how to choose these constants for general problems, the reader is referred to [63], where a perturbation analysis of regularization is performed. In order to solve the PMM subproblem, we apply one (or a few) iterations of an infeasible IPM. To that end, we alter the previous penalty function, by including logarithmic barriers, that is

$$\mathcal{L}^{IP-PMM}_{\rho_k,\delta_k}(x;\zeta_k,\eta_k) = \mathcal{L}^{PMM}_{\rho_k,\delta_k}(x;\zeta_k,\eta_k) - \mu_k\sum_{j\in\mathcal{I}}\ln x^j, \tag{2.6}$$

where $\mu_k > 0$ is the barrier parameter. In order to form the optimality conditions of this subproblem, we equate the gradient of $\mathcal{L}^{IP-PMM}_{\rho_k,\delta_k}$ with respect to x to the zero vector, i.e.,

$$\nabla f(x) - A^\top\eta_k + \frac{1}{\delta_k}A^\top(Ax-b) + \rho_k(x-\zeta_k) - \mathscr{P}^\top\begin{bmatrix}0_{|\mathcal{F}|}\\ \mu_k(X^{\mathcal{I}})^{-1}e_{|\mathcal{I}|}\end{bmatrix} = 0_n,$$

where $\mathscr{P}$ is an appropriate permutation matrix, such that $\mathscr{P}x_k = [(x_k^{\mathcal{F}})^\top,(x_k^{\mathcal{I}})^\top]^\top$. Next, we define the variables $y = \eta_k - \frac{1}{\delta_k}(Ax-b)$ and $z\in\mathbb{R}^n$, such that $z^{\mathcal{I}} = \mu_k(X^{\mathcal{I}})^{-1}e_{|\mathcal{I}|}$, $z^{\mathcal{F}} = 0$, to obtain the following (equivalent) system of equations:

$$\begin{bmatrix}\nabla f(x) - A^\top y - z + \rho_k(x-\zeta_k)\\ Ax + \delta_k(y-\eta_k) - b\\ X^{\mathcal{I}}z^{\mathcal{I}} - \mu_k e_{|\mathcal{I}|}\end{bmatrix} = \begin{bmatrix}0_n\\0_m\\0_{|\mathcal{I}|}\end{bmatrix}.$$

To approximately solve the previous mildly nonlinear system of equations, at every iteration k, we employ a damped perturbed Newton method (that is, we alter its right-hand side using a centering parameter $\sigma_k\in(0,1)$). In other words, at every iteration of IP-PMM we have available an iteration triple (x_k,y_k,z_k) and we want to solve the following system of equations:

$$\begin{bmatrix}-(\nabla^2 f(x_k)+\rho_k I_n) & A^\top & I_n\\ A & \delta_k I_m & 0_{m,n}\\ Z_k & 0_{n,m} & X_k\end{bmatrix}\begin{bmatrix}\Delta x\\ \Delta y\\ \mathscr{P}^\top\begin{bmatrix}0_{|\mathcal{F}|}\\ \Delta z^{\mathcal{I}}\end{bmatrix}\end{bmatrix} = \begin{bmatrix}\nabla f(x_k) - A^\top y_k + \sigma_k\rho_k(x_k-\zeta_k) - z_k\\ b - Ax_k - \sigma_k\delta_k(y_k-\eta_k)\\ \mathscr{P}^\top\begin{bmatrix}0_{|\mathcal{F}|}\\ \sigma_k\mu_k e_{|\mathcal{I}|} - X_k^{\mathcal{I}}z_k^{\mathcal{I}}\end{bmatrix}\end{bmatrix}, \tag{2.7}$$

Notice that all penalty parameters in the right-hand side are multiplied by σ_k. In essence σ_k determines how fast (or slow) these parameters are going to decrease in the next IP-PMM iteration. Following a standard development of IPMs, $x_k^{\mathcal{I}}$ is maintained positive and hence the logarithmic barrier in (2.6) is well defined while also $z_k^{\mathcal{I}}$ remains positive. From the third block-equation of (2.7) we have $\Delta z^{\mathcal{F}} = 0$ and

$$\Delta z^{\mathcal{I}} = (X_k^{\mathcal{I}})^{-1}(-Z_k^{\mathcal{I}}\Delta x^{\mathcal{I}} + \sigma_k\mu_k e_{|\mathcal{I}|} - X_k^{\mathcal{I}} z_k^{\mathcal{I}}).$$

In light of the previous computations, (2.7) reduces to:

$$\begin{bmatrix} -(\nabla^2 f(x_k) + \Xi_k + \rho_k I_n) & A^\top \\ A & \delta_k I_m \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} r_{1,k} \\ r_{2,k} \end{bmatrix}, \tag{2.8}$$

where

$$\begin{bmatrix} r_{1,k} \\ r_{2,k} \end{bmatrix} = \begin{bmatrix} \nabla f(x_k) - A^\top y_k + \sigma_k\rho_k(x_k - \zeta_k) - \mathscr{P}^\top \begin{bmatrix} 0_{|\mathcal{F}|} \\ \sigma_k\mu_k (X_k^{\mathcal{I}})^{-1} e_{|\mathcal{I}|} \end{bmatrix} \\ b - Ax_k - \sigma_k\delta_k(y_k - \eta_k) \end{bmatrix}, \tag{2.9}$$

and

$$\Xi_k \coloneqq \mathscr{P}^\top \begin{bmatrix} 0_{|\mathcal{F}|,|\mathcal{F}|} & 0_{|\mathcal{I}|,|\mathcal{F}|} \\ 0_{|\mathcal{F}|,|\mathcal{I}|} & (X_k^{\mathcal{I}})^{-1}(Z_k^{\mathcal{I}}) \end{bmatrix} \mathscr{P}.$$

For the rest of this paper, we will make use of the notation Ξ_k in cases where only a subset of the primal variables x are constrained to be non-negative. In the case where all the entries of x must satisfy this constraint, we will employ the standard IPM notation $\Theta_k \equiv \Xi_k^{-1}$, since in this case $\mathcal{F} = \emptyset$. In the special case where $\nabla^2 f(x_k)$ is a diagonal (or zero) matrix, it could be beneficial to further reduce system (2.8), by eliminating variables Δx. The resulting normal equations yield a positive definite system of equations that reads as follows:

$$\begin{aligned} &\left(A(\nabla^2 f(x_k) + \Xi_k + \rho_k I_n)^{-1} A^\top + \delta_k I_m\right) \Delta y \\ &= r_{2,k} + A(\nabla^2 f(x_k) + \Xi_k + \rho_k I_n)^{-1}(r_{1,k}). \end{aligned} \tag{2.10}$$

The parameters η_k, ζ_k are tuned as in [64]. In particular, we set $\eta_0 = y_0$ and $\zeta_0 = x_0$, where (x_0, y_0, z_0) is the starting point of IP-PMM. Then, at the end of every iteration k, we set $(\zeta_{k+1}, \eta_{k+1}) = (x_{k+1}, y_{k+1})$ only if the primal and dual residuals are decreased sufficiently. If the latter is not the case, we set $(\zeta_{k+1}, \eta_{k+1}) = (\zeta_k, \eta_k)$.

It has been demonstrated in [64] that IP-PMM using a single Newton step per iteration converges to an ϵ-optimal solution in a number of iterations that is polynomial with respect to the problem size n, if f is a convex quadratic function. Furthermore, the latter holds for linear semidefinite programming problems, even if one solves the Newton system inexactly, i.e., requiring only the residual to be bounded by a suitable multiple of the barrier parameter μ_k (see [65]). Nevertheless, the previous is not proven to hold for the general convex (nonlinear) case. In the latter case, one would have to employ Newton method combined with a *line-search* or a *trust-region* strategy (see, e.g., [3, 77]), in order to guarantee the convergence of the method. In all the cases analyzed in this work we make use of a simple Mehrotra-type [56] predictor-corrector scheme, which in general is sufficient to produce good directions that allow the method to converge quickly to the optimal solution. In the corrector stage, the right-hand side is approximated by a linearization of the function that is being minimized (see [72]).

2.3. Testing environment. The various specializations of IP-PMM discussed in the following sections have been implemented in MATLAB and compared with MATLAB implementations of state-of-the-art methods for each specific problem. All the tests were run with MATLAB R2019b on an Intel Xeon Platinum 8168 CPU with 192 GB of RAM, available from the *magicbox* server at the Department of Mathematics and Physics of the University of Campania "L. Vanvitelli".

3. Portfolio Selection Problem. Portfolio selection is one of the most central topics in modern financial economics. It deals with the decision problem of how to allocate resources among several competing assets in accordance with the investment objectives. For medium- and long-time horizons, the multi-period strategy is suitable, because it allows the change of the allocation of the capital across the assets, taking into account the evolution of the available information. In a multi-period setting, the investment period is partitioned into m sub-periods, delimited by $m+1$ rebalancing dates t^j. The decisions are taken at the beginning of each sub-period $[t^j, t^{j+1})$, $j = 1, ..., m$, and kept within it. The optimal portfolio is defined by the vector

$$w = [w_1^\top, w_2^\top, \dots, w_m^\top]^\top,$$

where $w_j \in \mathbb{R}^s$ is the portfolio of holdings at time t^j and s is the number of assets.

The mean-variance formulation proposed by Markowitz [55] was extended to a multi-period portfolio selection by Li and Ng [51], and in recent years there has been a significant advancement of both theory and methodologies. In a multi-period mean-variance framework, we fix a final target expected return and adopt as risk measure the function obtained by summing the single-period variance terms [19]:

$$\rho(w) = \sum_{j=1}^{m} w_j^\top C_j w_j,$$

where $C_j \in \mathbb{R}^{s\times s}$ is the covariance matrix, assumed to be positive definite, estimated at t^j. A common strategy to estimate Markowitz model parameters is to use historical data as predictive of the future behavior of asset returns. Different regularization techniques have been proposed to deal with ill-conditioning due to asset correlation; in the last years the ℓ_1-regularization has been used to promote sparsity in the solution [25]. It allows investors to reduce the number of positions to be monitored and held and the overall transaction costs. Another useful interpretation of the ℓ_1 norm is related to the amount of short positions (i.e., negative components in the solution), which indicate an investment strategy where an investor is selling borrowed stocks in the open market, expecting that the market will drop, in order to realize a profit. A suitable tuning of the regularization parameter permits short controlling in both the single- and the multi-period case [22, 25]. However, in the multi-period case, the sparsity in the solution does not guarantee the control of the transaction costs, especially if the pattern of the active positions (i.e., positive components in the solution) completely changes across periods. In this case, sparsity must be introduced in the variation, e.g., by adding an ℓ_1 term involving the differences of the wealth values allocated on the assets between two contiguous rebalancing times. This acts as a penalty on the portfolio turnover, which has the effect of reducing the number of transactions and hence the transaction costs [23, 28].

Thus, we consider the following fused lasso optimization problem for multi-period

portfolio selection [28]:

$$\begin{array}{ll} \min\limits_{w} & \frac{1}{2}w^\top Cw + \tau_1\|w\|_1 + \tau_2\|Lw\|_1, \\ \text{s.t.} & w_1^\top e_s = \xi_{\text{init}}, \\ & w_j^\top e_s = (e_s + r_{j-1})^\top w_{j-1}, \quad j = 2, \ldots, m, \\ & (e_s + r_m)^\top w_m = \xi_{\text{term}}, \end{array} \tag{3.1}$$

where $n = m\,s$, $C = \text{diag}(C_1, C_2, \ldots, C_m) \in \mathbb{R}^{n\times n}$ is a block diagonal symmetric positive definite matrix, $\tau_1, \tau_2 > 0$, $L \in \mathbb{R}^{(n-s)\times n}$ is the discrete difference operator representing the fused lasso regularizer, $r_j \in \mathbb{R}^s$ is the expected return vector at time t^j, ξ_{init} is the initial wealth, and ξ_{term} is the target expected wealth resulting from the overall investment. The first constraint is the initial budget constraint. The strategy is assumed to be self-financing, as constraints from 2 to m establish; this means that the value of the portfolio changes only because the asset prices change. The $(m+1)$-st constraint defines the expected final wealth. To deal with the non-separability of the objective function in (3.1), we introduce an auxiliary variable d, which is constrained to be equal to Lw, and we equivalently formulate problem (3.1) as follows:

$$\begin{array}{ll} \min\limits_{w,d} & \dfrac{1}{2}w^\top Cw + \tau_1\|w\|_1 + \tau_2\|d\|_1, \\ \text{s.t.} & \bar{A}w = \bar{b}, \\ & Lw = d, \end{array} \tag{3.2}$$

where the constraint matrix $\bar{A} \in \mathbb{R}^{(m+1)\times n}$ can be interpreted as an $(m+1)\times m$ lower bi-diagonal block matrix, with blocks of dimension $1 \times s$ defined by

$$\bar{A}^{i,j} = \begin{cases} e_s^\top & \text{if } i = j, \\ -(e_s + r_{i-1})^\top & \text{if } j = i+1, \\ 0_s^\top & \text{otherwise,} \end{cases}$$

and $\bar{b} = (\xi_{\text{init}}, 0, 0, ..., \xi_{\text{term}})^\top \in \mathbb{R}^{m+1}$.

3.1. Specialized IP-PMM for quadratic portfolio optimization problems. Using the standard trick described in Section 1, we split w and d into two vectors of the same size, representing the non-negative and non-positive parts of the entries of w and d respectively, i.e., $w = w^+ - w^-$ and $d = d^+ - d^-$. Then, problem (3.2) is reformulated as the following QP problem:

$$\min_{x} \ \frac{1}{2}x^\top Qx + c^\top x, \quad \text{s.t.} \ \ Ax = b, \ \ x \geq 0 \tag{3.3}$$

where we set $l = n - s$, $\overline{n} = 2(n+l) = 2s(2m-1)$, $\overline{m} = m+1+l = (m+1)+s(m-1)$,

$$x = [(w^+)^\top, (w^-)^\top, (d^+)^\top, (d^-)^\top]^\top \in \mathbb{R}^{\overline{n}},$$

$$Q = \begin{bmatrix} \begin{bmatrix} C & -C \\ -C & C \end{bmatrix} & 0_{2n,2l} \\ 0_{2l,2n} & 0_{2l,2l} \end{bmatrix} \in \mathbb{R}^{\overline{n}\times\overline{n}}, \quad A = \begin{bmatrix} \bar{A} & -\bar{A} & 0_{(m+1),2l} \\ L & -L & \begin{bmatrix} -I_l & I_l \end{bmatrix} \end{bmatrix} \in \mathbb{R}^{\overline{m}\times\overline{n}}, \tag{3.4}$$

$$c = [\tau_1, \ldots, \tau_1, \tau_2, \ldots, \tau_2]^\top \in \mathbb{R}^{\overline{n}}, \quad b = [\bar{b}^1, \ldots, \bar{b}^{m+1}, 0, \ldots, 0]^\top \in \mathbb{R}^{\overline{m}}.$$

3.1.1. Dropping Primal Variables. The optimal solution of problem (3.2) is expected to be sparse. On the other hand, in light of the reformulation given in (3.3), we anticipate (and verify in practice) that most of the primal variables w attain a zero value close to optimality. Such variables may significantly contribute to the ill conditioning of the matrix in (2.8) (see, e.g., [27, 37] and the references therein). In order to take advantage of this special property displayed by problem (3.3), we employ the following heuristic method, which aims at dropping variables x^j which are sufficiently close zero, their seemingly optimal value. This results in better conditioning of the augmented system, whose dimension is also significantly reduced close to optimality, thus decreasing the computational cost of each IPM iteration. In other words, as IP-PMM progresses, we project the problem onto a smaller space. After this reduced problem is solved, its optimal solution is expanded back to the original space by filling all earlier eliminated variables with zeros. This delivers an optimal solution to the original problem.

In particular, we set a threshold value $\epsilon_{\text{drop}} > 0$, and a large constant $\xi > 0$. At iteration $k = 0$, we define a set $\mathcal{V} = \emptyset$. Then, at every iteration k of IP-PMM, we check the following condition, for every $j \in \mathcal{I} \setminus \mathcal{V}$:

$$x_k^j \leq \epsilon_{\text{drop}} \quad \text{and} \quad z_k^j \geq \xi \cdot \epsilon_{\text{drop}} \quad \text{and} \quad (r_d)_k^j \leq \epsilon_{\text{drop}}, \tag{3.5}$$

where $(r_d)_k^j = (c - A^\top y_k + Q x_k - z_k)^j$ represents the dual infeasibility corresponding to the j-th variable. Any variable that satisfies the latter condition is dropped, that is, we set $x_k^j = 0$, $\mathcal{V} = \mathcal{V} \cup \{j\}$, $\mathcal{G} = \mathcal{F} \cup (\mathcal{I} \setminus \mathcal{V})$, we drop z_k^j and solve

$$\begin{bmatrix} -(Q^{\mathcal{G},\mathcal{G}} + \Xi_k^{\mathcal{G},\mathcal{G}} + \rho_k I_{|\mathcal{G}|}) & (A^{\mathcal{H},\mathcal{G}})^\top \\ A^{\mathcal{H},\mathcal{G}} & \delta_k I_{\overline{m}} \end{bmatrix} \begin{bmatrix} \Delta x^{\mathcal{G}} \\ \Delta y \end{bmatrix} = \begin{bmatrix} r_{1,k}^{\mathcal{G}} \\ r_{2,k} \end{bmatrix}, \tag{3.6}$$

where, Ξ_k is defined as in Section 2.2, $r_{1,k}$, $r_{2,k}$ are defined in (2.9) (by substituting $A^{\mathcal{H},\mathcal{G}}$ as the constraint matrix), and $\mathcal{H} = \{1, \ldots, \overline{m}\}$. We should note that this is a heuristic, since once a variable is dropped, it is not considered again until the method converges. Hence, one has to make sure that none of the nonzero variables x_k^j is dropped. Nevertheless, at the end of the optimization process we can test whether the variables in $\mathcal{V}$ are indeed nonzero. More specifically, once an optimal solution (x^*, y^*, z^*) is found, we compute:

$$z^{\mathcal{V}} = c^{\mathcal{V}} - (A^{\mathcal{H},\mathcal{V}})^\top y^* + Q^{\mathcal{V},\mathcal{G}}(x^*)^{\mathcal{G}}.$$

If there exists j such that $(z^{\mathcal{V}})^j \leq 0$, then we would identify that a variable x^j was incorrectly dropped. Otherwise, the optimal solution of the reduced problem coincides with the nonzero part of the optimal solution of (3.3). We notice that this methodology is not new. In particular, a similar strategy was employed in [39], where a special class of linear programming problems was solved using a primal-dual logarithmic barrier method.

3.2. Computational Experience. We test the effectiveness of the IP-PMM applied to the fused lasso model on the following real-world data sets:

1. FF48-FF100 (Fama & French 48-100 Industry portfolios, USA), containing 48-100 portfolios considered as assets, from July 1926 to December 2015.
2. ES50 (EURO STOXX 50), containing 50 stocks from 9 Eurozone countries (Belgium, Finland, France, Germany, Ireland, Italy, Luxembourg, the Netherlands and Spain), from January 2008 to December 2013.

3. FTSE100 (Financial Times Stock Exchange, UK), containing 100 assets, from July 2002 to April 2016.
4. SP500 (Standard & Poors, USA), containing 500 assets, from January 2008 to December 2016.
5. NASDAQC (National Association of Securities Dealers Automated Quotation Composite, USA), containing almost all stocks listed on the Nasdaq stock market, from February 2003 to April 2016.

Following [23, 24], we generate 10 problems with annual or quarterly rebalancing, after a preprocessing procedure that eliminates the elements with the highest volatilities. A rolling window (RW) for setting up the model parameters is considered. For each dataset, the length of the RWs is fixed in order to build positive definite covariance matrices and ensure statistical significance. Different datasets require different lengths for the RWs. In Table 1 we summarize the information on the test problems.

TABLE 1
Characteristics of the portfolio test problems (y = years, m = months)

Problem	**Assets**	**RW**	**Sub-periods**	$\overline{n}$
FF48-10	48	5 y	10 y	1632
FF48-20	48	5 y	20 y	3552
FF48-30	48	5 y	30 y	5472
FF100-10	96	10 y	10 y	3264
FF100-20	96	10 y	20 y	7104
FF100-30	96	10 y	30 y	10,944
ES50	50	1 y	22 m	4300
FTSE100	83	1 y	10 y	3154
SP500	431	2 y	8 y	11,206
NASDAQC	1203	1 y	10 y	45,714

We introduce some measures to evaluate the goodness of the optimal portfolios versus the benchmark one, in terms of risk, sparsity and transaction costs. We consider as benchmark the multi-period naive portfolio, based on the strategy for which at each rebalancing date the total wealth is equally divided among the assets. We assume that the investor has one unit of wealth at the beginning of the planning horizon, i.e., $\xi_{\text{init}} = 1$, and we set as expected final wealth the one provided by the benchmark. As in [23], we define:

$$ratio = \frac{w_{naive}^\top C w_{naive}}{w_{opt}^\top C w_{opt}}, \tag{3.7}$$

where w_{naive} and w_{opt} are respectively the naive portfolio and the optimal one. This value measures the risk reduction factor with respect to the benchmark. We consider the number of active positions as a measure of holding costs; then the value

$$ratio_h = \frac{\#\text{ active positions of } w_{naive}}{\#\text{ active positions of } w_{opt}} \tag{3.8}$$

measures the reduction factor of the holding costs with respect to the benchmark. Finally, we consider the number of variations in the weights as a measure of transaction costs. More precisely, if $w_j^i \neq w_{j+1}^i$ we assume that security i has been bought or sold in the period $[t^j, t^{j+1})$. Then we estimate the number of transactions as:

$$\mathcal{T} = trace(V^\top V),$$

where $V \in \mathbb{R}^{s\times(m-1)}$, with

$$v^{ij} = \begin{cases} 1 & \text{if } |w_j^i - w_{j+1}^i| \geq \epsilon, \\ 0 & \text{otherwise.} \end{cases}$$

and $\epsilon > 0$, in order to make sense in financial terms. A measure of the transaction reduction factor with respect to the benchmark is given by

$$ratio_t = \frac{\mathcal{T}_{naive}}{\mathcal{T}_{opt}}. \tag{3.9}$$

We consider a version of the presented IP-PMM algorithm in which the solution of problem (3.6) is computed by means of factorization, the parameter ϵ_{drop} controlling the heuristic described in Section 3.1.1 is set to 10^{-4}, and the constant ξ, which is used to ensure that the respective dual slack variable is bounded away from zero, is set to 10^2. We compare IP-PMM with the Split Bregman method, which is known to be very efficient for this kind of problems. In detail, we consider the Alternating Split Bregman algorithm used in [24], based on a further reformulation of problem (3.2) as

$$\begin{aligned} \min_{w,u,d} \quad & \frac{1}{2} w^\top C w + \tau_1 \|u\|_1 + \tau_2 \|d\|_1, \\ \text{s.t.} \quad & \bar{A} w = \bar{b}, \\ & Lw = d, \\ & w = u. \end{aligned}$$

This algorithm splits the minimization in three parts. Given w_k, u_k, d_k, the $(k+1)$-st iteration consists in the minimization of a quadratic function to determine w_{k+1} and the application of the soft-thresholding operator

$$[\mathcal{S}(v,\gamma)]^i = \text{sign}(v^i) \cdot \max(|v^i| - \gamma, 0),$$

where v is a real vector and $\gamma > 0$, to determine u_{k+1} and d_{k+1}. The optimal value w_{k+1} can be obtained by solving the system $Hw = p_{k+1}$, with

$$H = C + \lambda_1 \bar{A}^\top \bar{A} + \lambda_2 L^\top L + \lambda_3 I, \tag{3.10}$$

where $\lambda_1, \lambda_2, \lambda_3 > 0$ are fixed and p_{k+1} depends on the iteration. Since H is independent of the iteration and is symmetric positive definite, sparse, and banded, in [24] the authors compute its sparse Cholesky factorization only once and solve two triangular systems at each iteration. We refer to this algorithm as ASB-Chol.

The values of τ_1 and τ_2 in (3.1) are selected to guarantee reasonable portfolios in terms of short positions. We recall that from the financial point of view, negative solutions correspond to transactions in which an investor sells borrowed securities in anticipation of a price decline. In our runs we consider the smallest values of τ_1 and τ_2 that produce at most 2% of short positions. We set $\tau_1 = \tau_2 = 10^{-2}$ for the FF48 and FF100 data sets, $\tau_1 = \tau_2 = 10^{-3}$ for ES50 and SP500, $\tau_1 = 10^{-2}$ and $\tau_2 = 10^{-3}$ for FTSE, and $\tau_1 = 10^{-2}$ and $\tau_2 = 10^{-4}$ for NASDAQC.

In Table 2 we present the results obtained with IP-PMM and ASB-Chol on the test problems. The termination criteria of IP-PMM are the same as in [64], i.e., based on the relative reduction of the primal infeasibility $||Ax - b||$ (i.e. the constraints violation), the dual infeasibility $||\nabla f(x) - A^\top y - z||$, as well as complementarity (which is controlled by μ). The stopping criterion for ASB-Chol is based only on the relative

TABLE 2
IP-PMM vs ASB-Chol

	IP-PMM				
Problem	**Time (s)**	**Iters**	*ratio*	$ratio_h$	$ratio_t$
FF48-10	1.37e−1	12	2.32e+0	6.67e+0	1.66e+1
FF48-20	3.77e−1	16	2.28e+0	6.58e+0	2.13e+1
FF48-30	8.43e−1	21	4.64e+0	6.15e+0	1.69e+1
FF100-10	4.92e−1	12	1.58e+0	1.78e+1	4.36e+1
FF100-20	1.63e+0	15	1.81e+0	2.04e+1	4.92e+1
FF100-30	3.93e+0	21	5.82e+0	1.34e+1	3.60e+1
ES50	4.59e−1	14	2.12e+0	4.42e+0	5.75e+1
FTSE100	4.64e−1	14	1.85e+0	5.37e+1	6.09e+1
SP500	3.43e+1	16	1.57e+0	8.62e+1	1.50e+2
NASDAQC	7.05e+2	20	3.15e+0	2.73e+0	3.89e+2
	ASB-Chol				
Problem	**Time (s)**	**Iters**	*ratio*	$ratio_h$	$ratio_t$
FF48-10	1.67e−1	1431	2.33e+0	6.67e+0	1.66e+1
FF48-20	3.72e−1	1985	2.31e+0	7.93e+0	2.09e+1
FF48-30	1.12e+0	4125	4.64e+0	6.08e+0	1.66e+1
FF100-10	8.49e−1	3087	1.58e+0	1.78e+1	4.36e+1
FF100-20	2.09e+0	3635	1.80e+0	1.78e+1	4.27e+1
FF100-30	8.54e+0	9043	5.83e+0	1.12e+1	2.97e+1
ES50	9.70e−1	4297	2.05e+0	2.94e+0	4.26e+1
FTSE100	4.29e−1	1749	1.80e+0	5.07e+1	5.71e+1
SP500	1.98e+1	3728	1.74e+0	6.16e+1	1.01e+2
NASDAQC	8.84e+2	14264	3.15e+0	2.73e+0	3.89e+2

reduction of the primal feasibility $||\bar{A}x - b||$, which is a standard choice in literature. The relative tolerance for the two algorithms is $tol = 10^{-6}$, which guarantees that the values of *ratio* differ by at most 10%, so that both algorithms produce comparable portfolios in terms of risk. We note that the solution computed by ASB-Chol is thresholded by setting to zero all the entries with absolute value not exceeding the same value of ϵ_{drop} used in the IP-PMM dropping strategy. The results show that the optimal portfolios computed by IP-PMM and ASB-Chol outperform the benchmark ones in terms of all the metrics. Concerning $ratio_h$ and $ratio_t$, IP-PMM is generally able to produce greater values than ASB-Chol, which indicates a higher sparsity in the solution found by IP-PMM. IP-PMM generally performs comparably or better than ASB-Chol in terms of elapsed time. Although ASB-Chol is faster than IP-PMM on SP500 by 14.5 seconds (42%), IP-PMM is able to reach a better solution in terms of sparsity. When applied to FF100-30 IP-PMM produces a portfolio associated with lower transaction costs and takes less than half of the time required by ABS-Chol. When applied to NASDAQC, which is the largest problem under consideration, the two methods reach comparable solutions in terms of all the evaluation metrics, but IP-PMM needs about 20% less time (179 seconds) than ASB-Chol. This suggests that the use of IP-PMM can be beneficial especially when solving high-dimensional problems.

4. Classification models for functional Magnetic Resonance Imaging data. The functional Magnetic Resonance Imaging (fMRI) technique measures brain

spatio-temporal activity via Blood-Oxygen-Level-Dependent (BOLD) signals. Starting from the assumption that neuronal activity is coupled with cerebral blood flow, fMRI signals have been used to identify regions associated with functions such as speaking, vision, movement, etc.. By analyzing the different oxigenation levels in specific areas of the brain of healthy and ill patients, in the last decades fMRI signals have been used to investigate the effect on the brain functionality of tumors, strokes, head and brain injuries and of cognitive disorders such as schizophrenia or Alzheimer's (see [31, 46, 57] and the references therein).

In an fMRI scan, voxels representing regions of the brain of a patient are recorded at different time instances. The temporal resolution is usually in the order of a few seconds, while the spatial resolution generally ranges from 4-5 mm (for some full brain analyses) to 1 mm (for analyses on specific brain regions), which may amount up to about a million voxels. Since fMRI experiments are conducted over groups of patients, the dimensionality of the data is further increased. Therefore, the interpretation of fMRI results requires the analysis of huge quantities of data. To this aim, machine learning techniques are being increasingly used in recent years, because of their capability of dealing with massive amounts of data, incorporating also a-priori information about the problems they are targeted to [4, 5, 31, 32, 41, 57, 66].

Here we focus on the problem of training a binary linear classifier to distinguish between different classes of patients (e.g., ill/healthy) or different kinds of stimuli (e.g., pleasant/unpleasant), and to get information about the most significant brain areas associated with the related neural activity. The two classes are identified by the labels -1 and 1. We assume that the training set consists of s_{-1} 3-dimensional (3d) scans in class -1 and s_1 3d scans in class 1, where each 3d scan is reshaped as a row vector of size $q = q_1 \times q_2 \times q_3$, and q_i is the number of voxels along the i-th coordinate direction of the domain covering the brain. All the scans are stored as rows of a matrix $D \in \mathbb{R}^{s \times q}$, where $s = s_{-1} + s_1$.

We use a square loss function with the aim of determining an unbiased hyperplane in $\mathbb{R}^q$ that can separate the patients in the two classes. This leads to a minimization problem of the form:

$$\min \frac{1}{2s} \|Dw - \hat{y}\|^2, \tag{4.1}$$

where $\hat{y}$ is a vector containing the labels associated with each scan. Notice that the use of the Euclidean loss is a standard practice in the literature for the classification of fMRI data (see, e.g., [5, 41, 42, 45, 53]). Nevertheless, it should be observed that different loss functions could be employed as well (e.g. see [68, 80]), potentially leading to better classification accuracy in certain cases.

Since the number of patients is usually much smaller than the size of a scan, i.e., $s \ll q$, problem (4.1) is strongly ill posed and thus requires regularization. Recently, significant attention has been given to regularization terms encouraging the presence of structured sparsity, where smoothly varying nonzero coefficients of the solution are associated with small contiguous regions of the brain. This is motivated by the possibility of obtaining more interpretable solutions than those corresponding to other regularizers that do not promote sparsity or lead to sparse solutions without any structure (see [5, 41, 53] and the references therein).

Structured sparsity can be promoted, e.g., by using a combination of ℓ_1 and anisotropic Total Variation (TV) terms [2], which can be regarded as a fused lasso

regularizer [73]. The regularized problem reads

$$\min_{w} \ \frac{1}{2s} \left\| Dw - \hat{y} \right\|^2 + \tau_1 \left\| w \right\|_1 + \tau_2 \left\| Lw \right\|_1, \tag{4.2}$$

where $\|Lw\|_1$ is the discrete anisotropic TV of w, i.e., $L = [L_x^\top \ L_y^\top \ L_z^\top]^\top \in \mathbb{R}^{l \times q}$ is the matrix representing first-order forward finite differences in the x, y, z-directions at each voxel. By penalizing the difference between each voxel and its neighbors in each direction, one enforces the weights of the classification hyperplane (which share the 3d structure of the scans) to assume similar values for contiguous regions of the brain, thus leading to identify whole regions of the brain involved in the decision process.

The previous problem can be reformulated by introducing the variables $u = Dw$ and $d = Lw$ and applying the splitting

$$w = w^+ - w^-, \ d = d^+ - d^-, \qquad (w^+, w^-, d^+, d^-) \geq (0_q, 0_q, 0_l, 0_l).$$

Let $m = l + s$ and $n = s + 2q + 2l$. Using the previous variables, (4.2) can be equivalently written as:

$$\begin{aligned} \min_{x} \ & \frac{1}{2} x^\top Q x + \ c^\top x, \\ \text{s.t.} \ & Ax = b, \\ & x_{\mathcal{I}} \geq 0, \ x_{\mathcal{F}} \text{ free}, \ \mathcal{I} = \{s+1, \ldots, n\}, \ \mathcal{F} = \{1, \ldots, s\}, \end{aligned} \tag{4.3}$$

where $b = 0_{s+l} \in \mathbb{R}^m$,

$$x = [u^\top, (w^+)^\top, (w^-)^\top, (d^+)^\top, (d^-)^\top]^\top, \quad c = [-\frac{\hat{y}^\top}{s}, \tau_1 e_w^\top, \tau_1 e_w^\top, \tau_2 e_d^\top, \tau_2 e_d^\top]^\top \in \mathbb{R}^n,$$

and

$$Q = \begin{bmatrix} \frac{1}{s} I_s & 0_{s,(n-s)} \\ 0_{(n-s),s} & 0_{(n-s),(n-s)} \end{bmatrix} \in \mathbb{R}^{n \times n}, \quad A = \begin{bmatrix} -I_s & D & -D & 0_{s,l} & 0_{s,l} \\ 0_{l,s} & L & -L & -I_l & I_l \end{bmatrix} \in \mathbb{R}^{m \times n}. \tag{4.4}$$

4.1. Specialized IP-PMM for Fused Lasso Least Squares. Notice that problem (4.3) is in the same form as (2.5). In what follows, we present a specialized inexact IP-PMM, suitable for solving unconstrained fused lasso least squares problems. The proposed specialized IP-PMM is characterized by the two following implementation details. Firstly, instead of factorizing system (2.8), we employ an iterative method (namely, the Preconditioned Conjugate Gradient (PCG) method [44]) to solve system (2.10). Secondly, as suggested in Section 3.1.1, we take advantage of the fact that the optimal solution of problem (4.3) is expected to be sparse, and use the heuristic approach that allows us to drop many of the variables of the problem, when the method is close to the optimal solution.

4.1.1. Solving the Newton System. We focus on solving the normal equations in (2.10). Let k denote an arbitrary iteration of IP-PMM. We re-write the matrix in (2.10) without using the succinct notation introduced earlier:

$$M_k = \begin{bmatrix} M_{1,k} & M_{2,k}^\top \\ M_{2,k} & M_{3,k} \end{bmatrix}, \tag{4.5}$$

where:

$$
\begin{aligned}
M_{1,k} &= \left((\tfrac{1}{s}+\rho_k)^{-1}+\delta_k\right) I_s + D\left((\Xi_{w^+,k}+\rho_k I_q)^{-1}+(\Xi_{w^-,k}+\rho_k I_q)^{-1}\right) D^\top, \\
M_{2,k} &= L(\Xi_{w^+}+\rho_k I_q)^{-1} D^\top + L(\Xi_{w^-,k}+\rho_k I_q)^{-1} D^\top, \\
M_{3,k} &= L\left((\Xi_{w^+,k}+\rho_k I_q)^{-1}+(\Xi_{w^-,k}+\rho_k I_q)^{-1}\right) L^\top \\
&\quad + \left((\Xi_{d^+,k}+\rho_k I_l)^{-1}+(\Xi_{d^-,k}+\rho_k I_l)^{-1}+\delta_k I_l\right),
\end{aligned} \tag{4.6}
$$

while

$$
\Xi_k^{\mathcal{I}} = \begin{bmatrix} \Xi_{w^+,k} & 0_{q,q} & 0_{q,l} & 0_{q,l} \\ 0_{q,q} & \Xi_{w^-,k} & 0_{q,l} & 0_{q,l} \\ 0_{l,q} & 0_{l,q} & \Xi_{d^+,k} & 0_{l,l} \\ 0_{l,q} & 0_{l,q} & 0_{l,l} & \Xi_{d^-,k} \end{bmatrix},
$$

and Ξ_k is defined as in Section 2.2.

Notice that the matrix D in (4.1) is dense, and hence we expect $M_{1,k}$ and $M_{2,k}$ in (4.6) to also be dense. On the other hand, $M_{3,k}$ remains sparse, and we know that $l \gg s$. As a consequence, the Cholesky factors of the matrix in (4.5) would inevitably contain dense blocks. Hence, it might be prohibitively expensive to compute such a decomposition. Instead, we solve the previous system using a PCG method. In order to do so efficiently, we must find an approximation for the coefficient matrix in (4.5). Given the fact that $M_{3,k}$ is sparse, while $M_{1,k}$ and $M_{2,k}$ are dense, we would like to find an approximation for the dense blocks. A possible approach would be to approximate D by a low-rank matrix. Instead, based on the assumption $l \gg s$, we can approximate M_k by the following block-diagonal preconditioner:

$$
P_k = \begin{bmatrix} M_{1,k} & 0_{s,l} \\ 0_{l,s} & M_{3,k} \end{bmatrix}, \text{ where, } P_k^{-1} = \begin{bmatrix} M_{1,k}^{-1} & 0_{s,l} \\ 0_{l,s} & M_{3,k}^{-1} \end{bmatrix}. \tag{4.7}
$$

We observe that $M_{3,k}$ is a matrix of the form $LRL^\top + B$, where R and B are positive definite diagonal matrices and L comes from stacking three first-order forward finite-difference operators. It is easy to check that $M_{3,k}$ has a 3×3 block structure in which the diagonal blocks are Symmetric Diagonally Dominant M-matrices (SDDM). One could hence build a diagonal preconditioner by exploiting specialized strategies recently developed for this class of matrices [17, 49, 62]. Nevertheless, we notice that $M_{3,k}$ does have a sparse Cholesky factor, due to the sparsity displayed in the discrete anisotropic TV matrix L, which in our experiments guaranteed good performance. On the other hand, the Cholesky factor of $M_{1,k}$ is dense. However, computation and storage of this dense factor is possible, as we only need to perform $O(s^3)$ operations, and store $O(s^2)$ elements.

Spectral Analysis. Let us now further support the choice of the preconditioner in (4.7) by performing a spectral analysis of the preconditioned system $R_k = P_k^{-1} M_k$. In the following, we write $\mathcal{A} \times \mathcal{B}$ to denote a vector space whose elements are vectors $[a^\top, b^\top]^\top$ with $a \in \mathcal{A}$ and $b \in \mathcal{B}$.

THEOREM 4.1. *Let $D \in \mathbb{R}^{s \times q}$ be the matrix in (4.1). Let also $M_k \in \mathbb{R}^{(s+l)\times(s+l)}$ be the matrix defined in (4.5) and P_k the preconditioner defined in (4.7). Then, the preconditioned matrix $R_k = P_k^{-1} M_k$ has $l - \mathrm{rank}(D)$ eigenvalues $\lambda = 1$, whose respective eigenvectors form a basis for $\{0_s\} \times \{Null(M_{2,k}^\top)\}$. All the remaining eigenvalues of the preconditioned matrix satisfy $\lambda \in (\chi, 1) \cup (1, 2)$, where $\chi = \frac{\delta_k \rho_k}{\sigma_{\max}^2(A) + \rho_k \delta_k}$, δ_k, ρ_k are the regularization parameters of IP-PMM and A is defined in (4.4).*

Proof. Let us consider the following generalized eigenproblem:

$$M_k p = \lambda P_k p.$$

We partition the eigenvector p as $p = [p_s^\top, p_l^\top]^\top$. Using (4.5), and (4.7), the eigenproblem can be written as:

$$\begin{aligned} p_s + M_{1,k}^{-1} M_{2,k}^\top p_l &= \lambda p_s \\ M_{3,k}^{-1} M_{2,k} p_s + p_l &= \lambda p_l. \end{aligned} \tag{4.8}$$

From (4.6) and (4.7) we know that $M_k \succ 0$ and $P_k \succ 0$ and hence $\lambda > 0$. Let us separate the analysis in two cases.

Case 1: $\lambda = \mathbf{1}$. This is the case for every p such that

$$p \in \mathcal{S}_k = \{0_s\} \times \{\mathrm{Null}(M_{2,k}^\top)\},$$

which trivially satisfies (4.8) for $\lambda = 1$. Upon noticing that

$$\dim(\mathcal{S}_k) = \dim(\mathrm{Null}(M_{2,k}^\top)) = l - \mathrm{rank}(M_{2,k}^\top),$$

we can conclude that $R_k = P_k^{-1} M_k$ has an eigenvalue $\lambda = 1$ of multiplicity $l - \mathrm{rank}(M_{2,k}^\top)$. The respective eigenvectors form a basis of $\mathcal{S}_k$. Notice also, from (4.6), that $\mathrm{rank}(M_{2,k}^\top) = \mathrm{rank}(D) \leq s$.

Case 2: $\lambda \neq \mathbf{1}$. There are exactly $s + \mathrm{rank}(D)$ such eigenvalues. In order to analyze this case, we have to consider two generalized eigenproblems. On the one hand, from the first block equation in (4.8), we have that:

$$p_s = \frac{1}{\lambda - 1} M_{1,k}^{-1} M_{2,k}^\top p_l,$$

and hence, substituting this in the second block equation of (4.8) gives the following generalized eigenproblem:

$$G_{l,k} p_l = \nu M_{3,k} p_l, \tag{4.9}$$

where $G_{l,k} = M_{2,k} M_{1,k}^{-1} M_{2,k}^\top$ and $\nu = (\lambda - 1)^2$. By assumption $\lambda \neq 1$, and hence $\lambda = \pm\sqrt{\nu} + 1$. However, we have $M_k \succ 0$ and hence $M_{3,k} - M_{2,k} M_{1,k}^{-1} M_{2,k}^\top \succ 0$. Let us assume that the maximum eigenvalue of $M_{3,k}^{-1} G_{l,k}$ is greater than or equal to 1, i.e., $\nu_{\max} \geq 1$. By substituting $\nu_{\max}$ in (4.9) and multiplying both sides of the previous inequality by $p_l^\top$, we get

$$p_l^\top G_{l,k}\, p_l = \nu_{\max}\, p_l^\top M_{3,k}\, p_l$$

and hence

$$p_l^\top (G_{l,k} - M_{3,k})\, p_l \geq 0.$$

The latter contradicts the fact that $M_{3,k} - G_{l,k} \succ 0$, and thus $\nu_{\max} < 1$. In other words, $\lambda \in (0,1) \cup (1,2)$.

Similarly, starting from the second block equation of (4.8), we get

$$p_l = \frac{1}{\lambda - 1} M_{3,k}^{-1} M_{2,k} p_s,$$

and substituting this in the first block equation in (4.8) yields

$$G_{s,k} p_s = \nu M_{1,k} p_s,$$

where $G_{s,k} = M_{2,k}^\top M_{3,k}^{-1} M_{2,k}$ and $\nu = (\lambda-1)^2$. As before, we have that $\lambda = \pm\sqrt{\nu}+1$. Using the fact that $M_{1,k} - M_{2,k}^\top M_{3,k}^{-1} M_{2,k} \succ 0$, we can mirror the previous analysis to conclude that $\nu_{\max} < 1$, and hence $\lambda \in (0,1) \cup (1,2)$.

Finally, notice that as long as the primal and dual regularization parameters of IP-PMM, ρ_k and δ_k respectively, are bounded away from zero, so are the eigenvalues of $R_k = P_k^{-1} M_k$, for every iteration k of the algorithm. In particular, we have that

$$\lambda_{\min}(R_k) \geq \frac{\lambda_{\min}(M_k)}{\lambda_{\max}(P_k)} \geq \frac{\delta_k \rho_k}{\sigma_{\max}^2(A) + \rho_k \delta_k} = \chi,$$

where we used the fact that $\lambda_{\min}(M_k) \geq \delta_k$ and $\lambda_{\max}(P_k) \leq \frac{\sigma_{\max}^2(A)}{\rho_k} + \delta_k$, where A is defined in (4.4). ∎

4.1.2. Dropping Primal Variables. The preconditioner (4.7) may be computed (and applied) very efficiently as we expect the Cholesky factor of $M_{3,k}$ to preserve sparsity and $M_{1,k} \in \mathbb{R}^{s\times s}$ to be relatively small (recall that $s \ll l$). We deduce from Theorem 4.1 that the preconditioner defined in (4.7) remains effective as long as the regularization parameters ρ_k and δ_k are not too small. However, to attain convergence of IP-PMM ρ_k and δ_k have to be reduced and then, due to the nature of IPMs, the matrix in (2.10) becomes increasingly ill conditioned as the method approaches the optimal solution. This implies that the preconditioner defined in (4.7) has only a limited applicability. In particular, this means that there is a limited scope for refining it and we may not be able to prevent degrading behaviour of PCG when IPM gets very close to the optimal solution.

However, we notice that the optimal solution of problem (4.2) is expected to be sparse. Like in the portfolio optimization problem, in light of the reformulation (4.3), we know that most of the primal variables x converge to zero. Close to optimality the presence of such variables would adversely affect the conditioning of the matrix in (2.10). To prevent that, we employ a heuristic similar to the one introduced in Section 3.1.1 which consists of eliminating variables which approach zero and have an associated Lagrange multiplier bounded away from zero. Given $\epsilon_{\text{drop}} > 0$, $\xi > 0$ and $\mathcal{V} = \emptyset$, at every iteration k of IP-PMM, we add to $\mathcal{V}$ each variable $j \in \mathcal{I} \setminus \mathcal{V}$ satisfying condition (3.5), and replace (2.10) with the reduced system

$$\left(A^{\mathcal{H},\mathcal{G}}\left(Q^{\mathcal{G},\mathcal{G}} + \Xi_k^{\mathcal{G},\mathcal{G}} + \rho_k I_{|\mathcal{G}|}\right)^{-1}(A^{\mathcal{H},\mathcal{G}})^\top + \delta_k I_m\right)\Delta y = r_{2,k} + A^{\mathcal{H},\mathcal{G}}\left(Q^{\mathcal{G},\mathcal{G}} + \Xi_k^{\mathcal{G},\mathcal{G}} + \rho_k I_{|\mathcal{G}|}\right)^{-1} r_{1,k}^{\mathcal{G}}, \tag{4.10}$$

where $\mathcal{H} = \{1,\dots,m\}$, $\mathcal{G} = \mathcal{F} \cup (\mathcal{I} \setminus \mathcal{V})$, and $r_{1,k}$, $r_{2,k}$ are defined in (2.9) (with constraint matrix $A^{\mathcal{H},\mathcal{G}}$).

4.2. Computational Experience. We consider a dataset consisting of fMRI scans for 16 male healthy US college students (age 20 to 25), with the aim of analyzing two active conditions: viewing unpleasant and pleasant images [59]. The preprocessed and registered data[1] consist of 1344 scans of size 122,128 voxels (only voxels with probability greater than 0.5 of being in the gray matter are considered), with 42 scans considered per subject and active condition (i.e., 84 scans per subject in total).

[1] available from https://github.com/lucabaldassarre/neurosparse

In order to assess the performance of the IP-PMM on this type of problems, we carry out a comparison with two state-of-the-art algorithms for the solution of problem (4.2):

- FISTA. As done for the tests in [5], problem (4.2) is reformulated as

$$\min_w \ \frac{1}{2s} \|Dw - \hat{y}\|^2 + \|\hat{L}w\|_1,$$

  where $\hat{L} = \begin{bmatrix} \tau_1 I_q & \tau_2 L^\top \end{bmatrix}^\top$, and solved by a version of FISTA [7] in which the proximal operator associated with $\|\hat{L}w\|_1$ is approximated by 10 steps of an inner FISTA cycle.
- ADMM. We consider the ADMM method [12] applied to the problem

$$\begin{array}{ll} \min\limits_{w,u,d} & \frac{1}{2s} \|Dw - \hat{y}\|^2 + \tau_1 \|u\|_1 + \tau_2 \|d\|_1, \\ \text{s.t.} & w - u = 0_q, \\ & Lw - d = 0_l, \end{array}$$

  in which the minimization of the quadratic function associated with the update of w is approximated by 10 steps of the CG algorithm.

In Table 3 we show the results obtained by applying the algorithms to the solution of the fMRI data classification problem. For each choice of the pair of regularization parameters (τ_1, τ_2), we report the average results obtained in a Leave-One-Subject-Out (LOSO) cross-validation test over the full dataset of patients. This consists in using the data concerning 1 patient as the validation set and the data concerning the remaining patients as the training set. Because of this setting, for each problem the size of w is $q = 122{,}128$, the number of rows of D is $s = 1260$, and the dimension of $d = Lw$ is $l = 339{,}553$.

By preliminary experiments the choice $\tau_1 = \tau_2$ appeared the most appropriate. Furthermore, for the IP-PMM, the parameters ϵ_{drop} and ξ controlling the heuristic described in Section 4.1.2 are set to 10^{-6} and 10^2, respectively. To perform a fair comparison between the three algorithms, we consider a stopping criterion based on the execution time, which, after some preliminary tests, is fixed to 30 minutes. The solution of the normal equations system (4.10) is computed by the MATLAB `pcg` function, for which we set the maximum number of iterations to 2000 and the tolerance as

$$tol = \begin{cases} 10^{-4} & \text{if } \|r_{y,k}\| < 1, \\ \max\left\{10^{-8}, \ \frac{10^{-4}}{\|r_{y,k}\|}\right\} & \text{otherwise,} \end{cases}$$

where $r_{y,k}$ is the right-hand side of equation (4.10).

For each algorithm tested, we report the mean and the standard deviation for three quality measures of the solution: *classification accuracy* (ACC), *solution density* (DEN) and *corrected pairwise overlap* (CORR OVR) (see [5, Section 2.3.3]). Let N_f be the number of folders in the cross validation setting and let w_i be a given approximate solution to the problem associated with the i-th folder. For each w_i we define the accuracy (ACC) as the percentage of test vectors correctly classified by the linear model identified by w_i. Given a vector $v \in \mathbb{R}^q$, we define $\mathcal{Z}(v)$ as the set of indices corresponding to the nonzero components in v and $\mathcal{D}(v) = |\mathcal{Z}(v)|/q$ as the density of v. Hence, for each w_i the density (DEN) is computed as $\mathcal{D}(w_i)$. Finally, given any pair of indices $i, j \in \{1, \ldots, N_f\}$, the corrected pairwise overlap is defined as

$$\mathcal{O}^c_{i,j} = \frac{|\mathcal{Z}(w_i) \cap \mathcal{Z}(w_j)| - E}{\max\{|\mathcal{Z}(w_i)|, |\mathcal{Z}(w_j)|\}},$$

where E is the expected overlap between the support of two random vectors with density equal to $\mathcal{D}(w_i)$ and $\mathcal{D}(w_j)$, respectively, which is given by $E = q\,\mathcal{D}(w_i)\,\mathcal{D}(w_j)$. We observe that the corrected pairwise overlap, which may be the less common in the field of machine learning, is meant to measure the "stability" of the voxel selection. The three metrics are computed after thresholding the solution, as in [5]: after sorting the entries by their increasing magnitude, we set to zero the entries contributing at most to 0.01% of the ℓ_1-norm of the solution.

TABLE 3
Comparison of IP-PMM, FISTA and ADMM in terms of the LOSO cross-validation scores

Algorithm	$\tau_1 = \tau_2$	**ACC**	**DEN**	**CORR OVR**
IP-PMM	10^{-2}	86.16 ± 7.11	20.56 ± 6.63	43.47 ± 9.09
	$5 \cdot 10^{-2}$	84.90 ± 4.80	3.77 ± 0.84	62.70 ± 10.39
	10^{-1}	82.29 ± 6.22	2.49 ± 0.34	82.60 ± 9.24
FISTA	10^{-2}	86.90 ± 5.01	88.97 ± 0.71	5.43 ± 0.43
	$5 \cdot 10^{-2}$	84.15 ± 5.92	19.36 ± 0.86	65.50 ± 2.68
	10^{-1}	81.62 ± 7.58	5.14 ± 0.44	80.44 ± 5.72
ADMM	10^{-2}	86.46 ± 6.91	98.70 ± 0.03	0.03 ± 0.01
	$5 \cdot 10^{-2}$	85.57 ± 5.37	97.97 ± 0.05	0.15 ± 0.04
	10^{-1}	82.07 ± 6.51	97.50 ± 0.19	0.26 ± 0.13

By looking at Table 3, one can see that IP-PMM appears to be generally better than the other algorithms in enforcing the structured sparsity of the solution, presenting a good level of sparsity and overlap. It is worth noting that, because of its definition, the corrected pairwise overlap tends to zero as the density goes towards 100%. Hence, for ADMM, which seems to be unable to enforce sparsity in the solution, the overlap is close to zero. As suggested in [5], one can evaluate the results in terms of the distance of the pair (ACC, CORR OVR) from the pair $(100, 100)$ (the smaller the distance, the better the results). For the tests reported in the table, we can see that the best scores are obtained by IP-PMM with regularization parameters $\tau_1 = \tau_2 = 10^{-1}$, for which the average accuracy is 82.3% and the corrected overlap is 82.6% with an average solution density of 2.5%.

To further evaluate the efficiency of IP-PMM in the solution of this class of problems, we compare its performance in terms of elapsed time against the performance of FISTA on the problem where the two methods reach the best scores, i.e., with $\tau_1 = \tau_2 = 10^{-1}$. For all the 16 instances of the LOSO cross validation, we store the current solution of each algorithm after every minute and, at the end of the execution, we compute the three quality measures for such intermediate solutions. The results are shown in Figure 1 in terms of history of the mean values (lines) together with their 95% confidence intervals (shaded regions). From the plots we can see that while FISTA reaches the measures reported in Table 3 at the end of the 30-minute run, the performance of IP-PMM stabilizes after about 20 minutes. At the 20 minutes mark we observe that for IP-PMM the value of each of the three metrics is the same as the one reported in Table 3. For FISTA, while the accuracy (81.32%) and overlap (80.54%) have similar values as those reported in the table, we observe a larger density (6.83%).

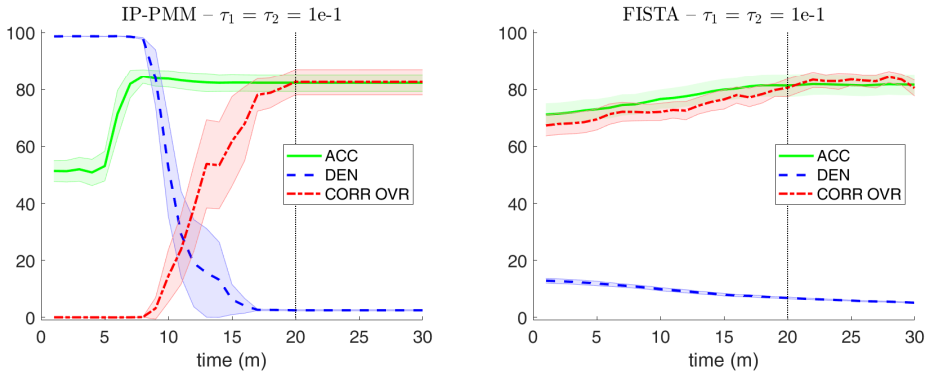

FIG. 1. *History of classification accuracy, solution density and corrected pairwise overlap for IP-PMM (left) and FISTA (right), in the case* $\tau_1 = \tau_2 = 10^{-1}$. *For the three quantities we report average measures with* 95% *confidence intervals.*

5. TV-based Poisson Image Restoration. Next we consider the restoration of images corrupted by Poisson noise, which arises in many applications, such as fluorescence microscopy, computed tomography (CT) and astronomical imaging (see, e.g., [29] and the references therein). In the discrete formulation of the restoration problem, the object to be restored is represented by a vector $w \in \mathbb{R}^n$ and the measured data are assumed to be a vector $g \in \mathbb{N}_0^m$, whose entries g^j are samples from m independent Poisson random variables G^j with probability

$$P(G^j = g^j) = \frac{e^{-(Dw+a)^j}\left[(Dw+a)^j\right]^{g^j}}{g^j!},$$

where $a \in \mathbb{R}_+^m$ models the background radiation detected by the sensors. The matrix $D = (d^{ij}) \in \mathbb{R}^{m\times n}$ models the functioning of the imaging system and satisfies

$$d^{ij} \geq 0 \text{ for all } i,j, \qquad \sum_{i=1}^{m} d^{ij} = 1 \text{ for all } j.$$

Here we assume that D represents a convolution operator with periodic boundary conditions, which implies that D has a Block-Circulant structure with Circulant Blocks (BCCB). Hence, Dw is computed expeditiously using the 2-dimensional Fast Fourier Transform (FFT). The maximum-likelihood approach [10] for the estimation of u leads to the minimization of the *Kullback-Leibler (KL) divergence* of $Dw + a$ from g:

$$D_{KL}(w) \equiv D_{KL}(Dw+a, g) = \sum_{j=1}^{m}\left(g^j \ln\frac{g^j}{(Dw+a)^j} + (Dw+a)^j - g^j\right), \tag{5.1}$$

where we set $g^j \ln(g^j/(Dw+a)^j) = 0$ if $g^j = 0$ (we implicitly assume that g has been converted into a real vector with entries ranging in the same interval as the entries of w). Since the estimation problem is highly ill conditioned, a regularization term is added to (5.1). We consider the Total Variation (TV) [67], which has received considerable attention because of its ability of preserving edges and smoothing flat areas of the images. Notice that, while it may introduce staircase artifacts, TV is still applied in many medical and biological applications (see, e.g., [6, 58, 83] and J. Huang's webpage[2]). The feasible set of the problem is defined by non-negativity

[2] http://ranger.uta.edu/~huang/R_CSMRI.htm

constraints on the image intensity and the linear constraint $\sum_{i=1}^{n} w^i = \sum_{j=1}^{m}(g^j - a^j) \equiv r$ which guarantees preservation of the total intensity of the image.

The resulting model is

$$\begin{array}{ll} \min\limits_{w} & D_{KL}(w) + \lambda \|Lw\|_1 \\ \text{s.t.} & e_n^\top w = r, \\ & w \geq 0, \end{array} \tag{5.2}$$

where $L \in \mathbb{R}^{l \times n}$ is the matrix arising from the discretization of the TV functional (as in [16]).

5.1. Specialized IP-PMM for Image Restoration Problems. By employing the splitting strategy used in the previous sections, we can transform problem (5.2) to the following equivalent form:

$$\begin{array}{ll} \min\limits_{x} & f(x) \equiv D_{KL}(w) + c^\top u, \\ \text{s.t.} & Ax = b, \\ & x \geq 0, \end{array} \tag{5.3}$$

where, after introducing the additional constraint $d = Lw$, and letting $\overline{m} = l + 1$, $\overline{n} = n + 2l$, we set $x = [w^\top,\ u^\top]^\top \in \mathbb{R}^{\overline{n}}$, $u = [(d^+)^\top,\ (d^-)^\top]^\top \in \mathbb{R}^{2l}$, $c = \lambda\, e_{2l}$, $b = [r,\ 0_l^\top]^\top \in \mathbb{R}^{\overline{m}}$, and

$$A = \begin{bmatrix} e_n^\top & 0_l^\top & 0_l^\top \\ L & -I_l & I_l \end{bmatrix} \in \mathbb{R}^{\overline{m} \times \overline{n}}.$$

We solve problem (5.3) by using IP-PMM combined with a perturbed composite Newton method [72]. Following the presentation in Section 2.2, we know that at the k-th iteration of the method we have to solve two linear systems of the form of (2.8). In order to avoid factorizations, every such system is solved using the preconditioned MINimal RESidual (MINRES) method [61]. In order to accelerate the convergence of MINRES, we employ a block-diagonal preconditioner, which uses a diagonal approximation of $\nabla^2 f(x)$. More specifically, at iteration k of IP-PMM, we have the following coefficient matrix:

$$M_k = \begin{bmatrix} -H_k & A^\top \\ A & \delta_k I_{\overline{m}} \end{bmatrix},$$

where $H_k = (\nabla^2 f(x_k) + \Theta_k^{-1} + \rho_k I_{\overline{n}})$, and we precondition it using the matrix

$$\widetilde{M}_k = \begin{bmatrix} \widetilde{H}_k & 0_{\overline{n},\overline{m}} \\ 0_{\overline{m},\overline{n}} & A\,\widetilde{H}_k^{-1} A^\top + \delta_k I_{\overline{m}} \end{bmatrix}, \tag{5.4}$$

where $\widetilde{H}_k$ is a diagonal approximation of H_k. In order to analyze the spectral properties of the preconditioned matrix, we follow the developments in [9]. More specifically, we define $\widehat{H}_k \coloneqq \widetilde{H}_k^{-\frac{1}{2}} H_k \widetilde{H}_k^{\frac{1}{2}}$, and let:

$$\alpha_H = \lambda_{\min}(\widehat{H}_k), \qquad \beta_H = \lambda_{\max}(\widehat{H}_k), \qquad \kappa_H = \frac{\beta_H}{\alpha_H}.$$

Using this notation, we know that an arbitrary element of the numerical range of this matrix is represented as $\gamma_H \in W(\widehat{H}_k) = [\alpha_H, \beta_H]$. Furthermore, we observe that in

the special case where $\widetilde{H}_k = \mathrm{diag}(H_k)$, we have $\alpha_H \leq 1 \leq \beta_H$ since

$$\frac{1}{n+2l}\sum_{i=1}^{n+2l}\lambda_i(\widehat{H}_k^{-1}H_{k,j}) = \frac{1}{n+2l}\,\mathrm{Tr}(\hat{H}_k^{-1}H_k) = 1.$$

THEOREM 5.1. *Let k be an arbitrary IP-PMM iteration. Then, the eigenvalues of $\widetilde{M}_k^{-1}M_k$ lie in the union of the following intervals:*

$$I_- = \left[-\beta_H - 1, -\alpha_H\right], \qquad I_+ = \left[\frac{1}{1+\beta_H}, 1\right].$$

Proof. The proof follows exactly the developments in [9, Theorem 3.3]. ∎

In problem (5.3), $f(x) = D_{KL}(w) + c^\top u$ and hence

$$\nabla f(x) = \begin{bmatrix}\nabla D_{KL}(w)\\ c\end{bmatrix}, \qquad \nabla^2 f(x) = \begin{bmatrix}\nabla^2 D_{KL}(w) & 0_{n,2l}\\ 0_{2l,n} & 0_{2l,2l}\end{bmatrix},$$

where

$$\nabla D_{KL}(w) = D^\top\left(e_m - \frac{g}{Dw+a}\right), \qquad \nabla^2 D_{KL}(w) = D^\top U(w)^2 D,$$

with $U(w) = \mathrm{diag}\left(\frac{\sqrt{g}}{Dw+a}\right)$. Here the ratios and the square root are assumed to be component-wise. Notice that D might be dense; however, as previously noted, its action can be computed via the FFT. Unfortunately, $D^\top U(w)^2 D$ is not expected to be close to multilevel circulant. Even if it could be well-approximated by a multilevel circulant matrix, the scaling matrix of IP-PMM would destroy this structure. In other words, we use the structure of D only when applying it to a vector. As a result, we only store the first column of D and we use the FFT to apply this matrix to a vector. This allows us to compute the action of the Hessian easily.

Remark 5.2. The obvious choice would be to employ the approximation $\widetilde{H}_k = \mathrm{diag}(H_k)$, but the structure of the problem makes this choice rather expensive. A more efficient alternative is to use $\widetilde{H}_k = U(w_k)^2$, which is easier to compute and, as we will see in the following section, leads to good reconstruction results in practice.

5.2. Computational Experience. To evaluate the performance of the IP-PMM on this class of problems, we consider a set of three 256×256 grayscale images, which are presented in Figure 2. For each of the three images we set up three restoration tests, where the images are corrupted by Poisson noise and D represents one of the following blurs: Gaussian blur (GB), motion blur (MB), and out-of-focus blur (OF) (see, e.g., [43] for further details).

We compare the proposed method with the state-of-the-art Primal-Dual Algorithm with Linesearch (PDAL) proposed in [54]. By following the example of [79, Algorithm 2], problem (5.2) is reformulated as

$$\min_{w}\max_{p,y}\; g^\top \ln(1+y) - y^\top(Dw+a) - \lambda\, w^\top L^\top p + \chi_\infty(p) + \chi_{\mathcal{C}}(w), \tag{5.5}$$

where χ_∞ is the characteristic function of the ∞-norm unit ball and $\chi_{\mathcal{C}}$ the characteristic function of the feasible set $\mathcal{C}$ of problem (5.2). It is worth noting that the PDAL algorithm for the solution of problem (5.5) requires at each step a projection

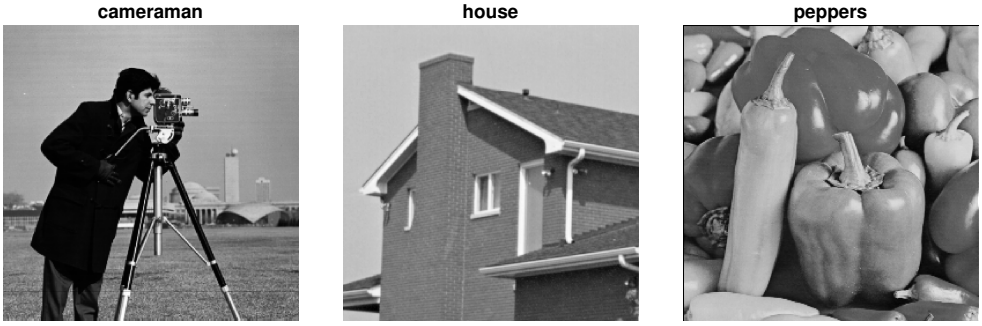

FIG. 2. *The three* 256×256 *grayscale images of the image restoration tests.*

on the feasible set $\mathcal{C}$, which is performed here by using the secant algorithm proposed by Dai and Fletcher in [26]. Concerning the parameters of PDAL, we use the same notation and tuning as in [54]. Following Section 6 of that paper, we set $\mu = 0.7$, $\delta = 0.99$ and $\beta = 25$. The initial steplength is $\tau = \sqrt{1/\omega}$, where ω is an estimate of $\|M^\top M\|$ and $M = \begin{bmatrix} D^\top & L^\top \end{bmatrix}^\top$ is the matrix linking the primal and dual variables. In the IP-PMM, we use the MINRES code by Michael Saunders and co-workers[3] for which we set the relative tolerance $tol = 10^{-4}$ and the maximum number of iterations at each call equal to 20. The regularization parameter λ is determined by trial and error to minimize the Root Mean Square Error (RMSE) obtained by IP-PMM. We recall that, denoting the original image as $\bar{w} \in \mathbb{R}^n$, for any given approximate solution $w \in \mathbb{R}^n$ we have that

$$\mathrm{RMSE}(w) = \frac{1}{\sqrt{n}} \|w - \bar{w}\|_2.$$

For all the problems, the starting point is chosen to be the noisy and blurry image, i.e., g.

For all 9 tests we run 20 iterations of the IP-PMM method and let PDAL run for the same amount of time. In Figure 3 we report a comparison between the two algorithms in terms of elapsed time versus Root Mean Square Error (RMSE) in the solution of the 9 instances described above. As can be seen from the plots, the IP-PMM clearly outperforms PDAL on the instances with GB and OF (columns 1 and 3, respectively, of Figure 3), while on the instances characterized by MB the two algorithms perform comparably.

To better analyze the difference between the solutions provided by the two algorithms, one can look at Table 4, where we report the value of three scores: RMSE, Peak Signal-to-Noise Ratio (PSNR), which is defined as

$$\mathrm{PSNR}(w) = 20 \log_{10} \frac{\max_i \bar{w}^i}{\mathrm{RMSE}(w)},$$

and Mean Structural SIMilarity (MSSIM), which is a structural similarity measure related to the perceived visual quality of the image (see [78] for a detailed definition). It is worth noting that for RMSE smaller values are better, while for PSNR and MSSIM, higher values indicate better noise removal and perceived similarity between the restored and original image, respectively. From the table it is clear that in all the considered cases IP-PMM is able to produce a better restored image than

[3] available from https://web.stanford.edu/group/SOL/software/minres/

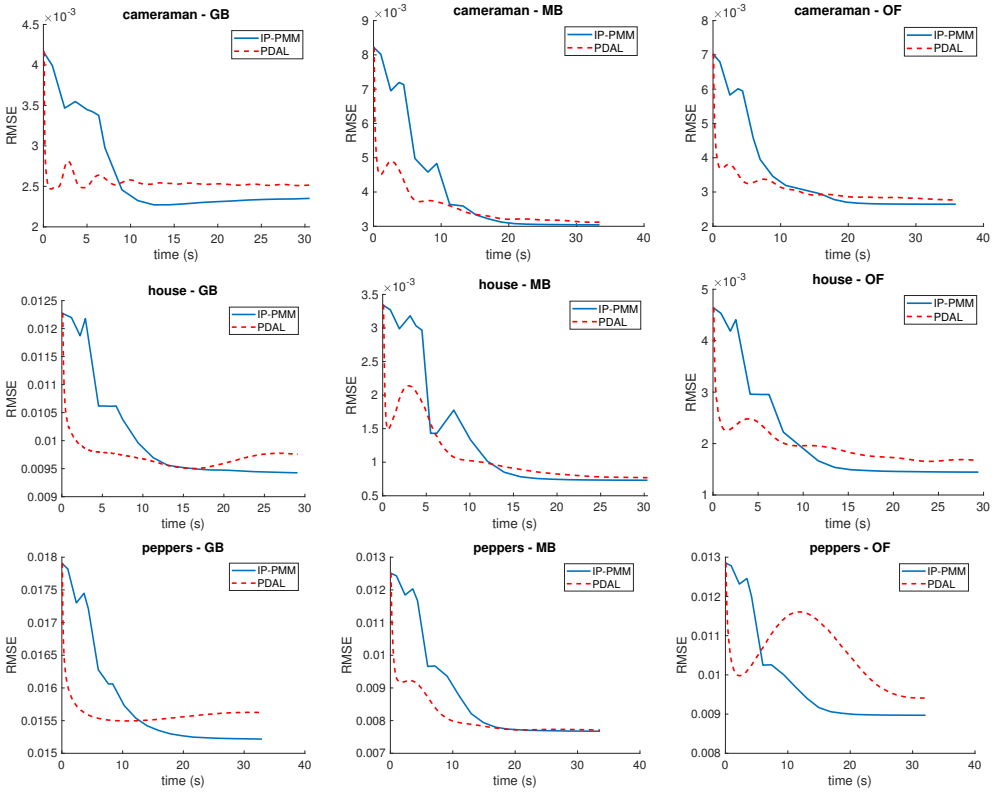

FIG. 3. *Comparison between IP-PMM and PDAL in terms of Root Mean Square Error (RMSE) vs execution time in the solution of the 9 image restoration problems. From top to bottom, the rows refer to the cameraman instances, the house instances and the peppers instances, respectively. From left to right, the columns refer to the GB, MB and OF, respectively.*

TABLE 4

Comparison between IP-PMM and PDAL in terms of RMSE, PSNR and MSSIM computed at the solutions provided by the two algorithms.

	IP-PMM			PDAL		
Problem	**RMSE**	**PSNR**	**MSSIM**	**RMSE**	**PSNR**	**MSSIM**
cameraman - GB	4.85e−2	2.63e+1	8.33e−1	5.02e−2	2.60e+1	8.22e−1
cameraman - MB	5.52e−2	2.52e+1	8.11e−1	5.59e−2	2.51e+1	7.77e−1
cameraman - OF	5.14e−2	2.58e+1	7.98e−1	5.26e−2	2.56e+1	7.62e−1
house - GB	9.71e−2	2.03e+1	7.51e−1	9.88e−2	2.01e+1	6.92e−1
house - MB	2.70e−2	3.14e+1	8.67e−1	2.77e−2	3.11e+1	8.43e−1
house - OF	3.80e−2	2.84e+1	8.33e−1	4.09e−2	2.78e+1	7.70e−1
peppers - GB	1.23e−1	1.82e+1	7.46e−1	1.25e−1	1.81e+1	6.57e−1
peppers - MB	8.76e−2	2.12e+1	8.90e−1	8.78e−2	2.11e+1	8.72e−1
peppers - OF	9.47e−2	2.05e+1	8.01e−1	9.70e−2	2.03e+1	6.60e−1

PDAL, having always a larger MSSIM, also when the RMSE and PSNR values are comparable.

For the sake of space, we now restrict the comparison to the cases where the two algorithms seem to have reached equivalent solutions in terms of RMSE, to understand

the differences in the restored images. We focus on the three instances in which D represents MB (second column of Figure 3). In Figure 4 we report the results for cameraman, house and peppers with MB. By looking at the images one can see that those reconstructed by IP-PMM appear to be smoother (look, for example, at the sky in cameraman and house), which somehow indicates that the IP-PMM is better than PDAL in enforcing the TV regularization. Observe that this "visual" difference is reflected by the higher values of MSSIM reported for IP-PMM in Table 4.

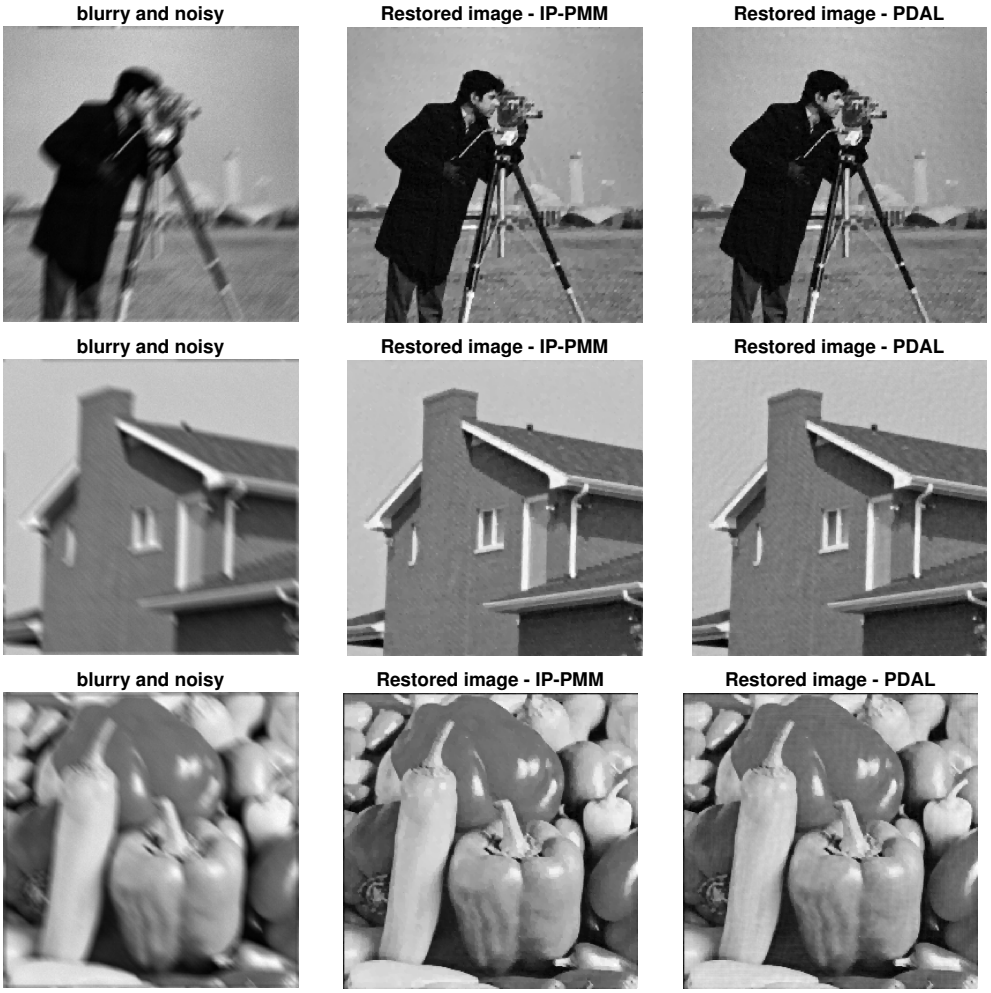

FIG. 4. *Results on cameraman, house and peppers with MB: noisy and blurry images (left), images restored by IP-PMM (center), images restored by PDAL (right).*

6. Linear Classification via Regularized Logistic Regression. Finally, we deal with the problem of training a linear binary classifier. Let us consider a matrix $D \in \mathbb{R}^{n\times s}$ whose rows $(d^i)^\top$, with $i \in \{1,\dots,n\}$, represent the training points, and a vector of labels $g \in \{-1, 1\}^n$. In other words, we have a training set with n binary-labeled samples and s features. According to the logistic model, the conditional

probability of having the label g^i given the point d_i has the form

$$p_{log}(w)_i = P(g^i|d^i) = \frac{1}{1+e^{-g^i\, w^\top d^i}},$$

where $w \in \mathbb{R}^s$ is the vector of weights determining the unbiased linear model under consideration. By following the maximum-likelihood approach, the weight vector w can be obtained by maximizing the log-likelihood function or, equivalently, by minimizing the *logistic loss function*, i.e., by solving

$$\min_w \ \phi(w) \equiv \frac{1}{n}\sum_{i=1}^{n}\phi_i(w), \qquad \phi_i(w) = \log\left(1+e^{-g^i\, w^\top d^i}\right).$$

To cope with the inherent ill-posedness of the estimation process, a regularization term is usually added to the previous model. For large-scale instances, where the features tend to be redundant, an ℓ^1-regularization term is usually introduced to enforce sparsity in the solution, thus embedding feature selection in the training process. This results in the well-studied ℓ^1-regularized logistic regression model:

$$\min_w \ \phi(w) + \tau\|w\|_1, \tag{6.1}$$

where $\tau > 0$.

As done in the previous sections, we can replace the nonsmooth model (6.1) with an equivalent smooth convex programming problem, i.e.,

$$\begin{aligned} &\min_x \ f(x) \equiv \phi(w) + c^\top u, \\ &\text{s.t. } Ax = b, \\ &\qquad u \geq 0, \end{aligned} \tag{6.2}$$

where, after introducing the additional constraint $u = w$, with $u = [(d^+)^\top,\ (d^-)^\top]^\top \in \mathbb{R}^{2s}$, and letting $\overline{m} = s$, $\overline{n} = 3s$, we set $x = [w^\top,\ u^\top]^\top \in \mathbb{R}^{\overline{n}}$, $c = \tau\, e_{2s}$, $b = 0_{\overline{m}}$, and $A \in \mathbb{R}^{\overline{m}\times\overline{n}}$ defined as $A = [I_s \quad -I_s \quad I_s]$. The version of IP-PMM solving problem (6.2) is very similar to the one used to solve (5.3). The only difference here lies in the preconditioner. In particular, when solving problems of the form (6.2), we use the preconditioner defined in (5.4) (and subsequently analyzed in Theorem 5.1), but we set $\widetilde{H}_k = \mathrm{diag}(H_k)$.

6.1. Computational Experience. To illustrate the performance of the IP-PMM on this class of problems, we consider a set of three linear classification problems from the LIBSVM dataset for binary classification[4]. The names of the datasets, together with their number of features, training points and testing points are summarized in Table 5. For real-sim there is no predetermined separation of data between train and test, hence we apply a hold-out strategy keeping 30% of the data for testing.

To overcome the absence of the hyperplane bias in model (6.1), we add to the data matrices a further column with all ones, hence the resulting size of the problems is equal to $s+1$. For all the problems we set $\tau = \frac{1}{n}$, which is a standard choice in the literature.

To assess the effectiveness and efficiency of the proposed method we compare it with two state-of-the-art methods:

[4]available from https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html

TABLE 5
Characteristics of the ℓ^1-regularized logistic regression problems

Problem	**Features**	**Train pts**	**Test pts**
gisette	5000	6000	1000
rcv1	47,236	20,242	677,399
real-sim	20,958	50,617	21,692

- an ADMM [12][5];
- a MATLAB implementation of the newGLMNET method [81] used in LIBSVM, developed by the authors of [82][6].

As in the tests presented in Section 5.2, the solution of the augmented system in IP-PMM is performed by means of the MINRES implementation by Michael Saunders' team, with maximum number of iterations equal to 20 and tolerance $tol = 10^{-4}$.

We compare the three algorithms in terms of objective function value and classification error versus execution time, on runs lasting 15 seconds. The plots are reported in Figure 5. The IP-PMM is comparable with newGLMNET on the gisette instance, characterized by a very dense ($> 99\%$) training data matrix, and both IP-PMM and newGLMNET clearly outperform ADMM. On the rcv1 and real-sim instances the IP-PMM method sightly outperforms newGLMNET in terms of classification error, and it is noticeably better in terms of the objective function value.

Remark 6.1. Let us notice that in the presented experiments, in order to invert each associated preconditioner, we needed to perform a Cholesky decomposition of an approximate normal equations matrix $A\widetilde{H}_k^{-1}A^\top + \delta_k I_{\overline{m}}$ (or a sub-matrix of it; e.g., as in Section 4), where $\widetilde{H}_k \approx \nabla^2 f(x_k) + \Theta_k^{-1} + \rho_k I_{\overline{n}}$. In certain cases, if A has full row-rank, one could instead employ an approximation based on a random sketching strategy, presented in [20, 21]. We should mention however, that this forces one to employ a singular value decomposition to invert the resulting matrix instead of a Cholesky decomposition (which is expected to be faster on the sparse problems under consideration). Furthermore, in the case of rank-deficient matrix A this would create certain computational issues, as then the dropping heuristic presented in Section 3.1.1 would be very expensive to employ (see the discussion in [20, Section 5]). Nevertheless, we should mention that in certain applications for which either most of the singular values of A are close to zero, or the Cholesky decomposition of the approximated Schur complement is expensive, such an approach could prove advantageous.

7. Conclusions. We have presented specialized IPMs for quadratic and general convex nonlinear optimization problems that model various sparse approximation instances. We have shown that by a proper choice of linear algebra solvers, which are a key issue in IPMs, we are able to efficiently solve the larger but smooth optimization problems coming from a standard reformulation of the original ones. This confirms the ability of IPMs to handle large sets of linear equality and non-negativity constraints. Computational experiments have been performed on diverse applications: multi-period portfolio selection, classification of fMRI data, restoration of blurry images corrupted by Poisson noise, and linear binary classification via regularized logistic regression. Comparisons with state-of-the-art first-order methods, which are widely used to tackle sparse approximation problems, have provided evidence that the pre-

[5] available from http://www.stanford.edu/~boyd/papers/distr_opt_stat_learning_admm.html
[6] available from https://github.com/ZiruiZhou/IRPN

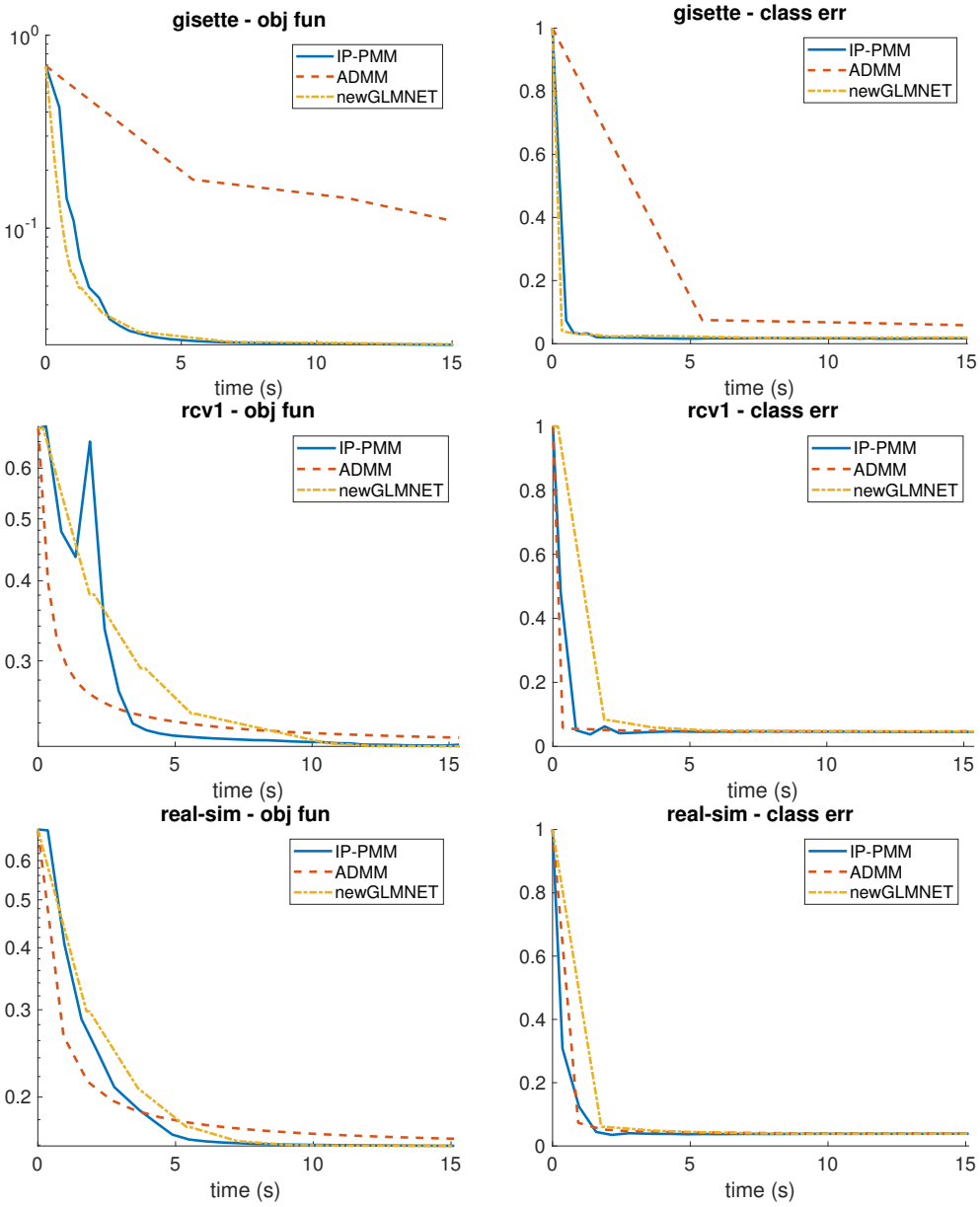

FIG. 5. *Results on the three ℓ^1-regularized logistic regression problems (objective function value and classification error versus execution time).*

sented IPM approach can offer a noticeable advantage over those methods, especially when dealing with not-so-well conditioned problems.

We also believe that the results presented in this work may provide a basis for an in-depth analysis of the application of IPMs to many sparse approximation problems, and we plan to work in that direction in the future.

Acknowledgments. We thank the anonymous reviewers for their careful reading of the manuscript and their insightful remarks and suggestions, which allowed us to improve the quality of our work.

REFERENCES

[1] A. Altman and J. Gondzio, *Regularized symmetric indefinite systems in interior point methods for linear and quadratic optimization*, Optimization Methods and Software, 11–12 (1999), pp. 275–302, https://doi.org/10.1080/10556789908805754.

[2] A. Argyriou, L. Baldassarre, C. A. Micchelli, and M. Pontil, *On sparsity inducing regularization methods for machine learning*, in Empirical Inference: Festschrift in Honor of Vladimir N. Vapnik, B. Schölkopf, Z. Luo, and V. Vovk, eds., Berlin, Heidelberg, 2013, Springer, pp. 205–216, https://doi.org/10.1007/978-3-642-41136-6_18.

[3] P. Armand and R. Omheni, *A mixed logarithmic barrier-augmented Lagrangian method for nonlinear optimization*, Journal of Optimization Theory and Applications, 173 (2017), pp. 523–547, https://doi.org/10.1007/s10957-017-1071-x.

[4] L. Baldassarre, J. Mourão-Miranda, and M. Pontil, *Structured sparsity models for brain decoding from fMRI data*, in 2012 Second International Workshop on Pattern Recognition in NeuroImaging, July 2012, pp. 5–8, https://doi.org/10.1109/PRNI.2012.31.

[5] L. Baldassarre, M. Pontil, and J. Mourão-Miranda, *Sparsity is better with stability: Combining accuracy and stability for model selection in brain decoding*, Frontiers in Neuroscience, 11 (2017), https://doi.org/10.3389/fnins.2017.00062.

[6] R. C. Barnard, H. Bilheux, T. Toops, E. Nafziger, C. Finney, D. Splitter, and R. Archibald, *Total variation-based neutron computed tomography*, Review of Scientific Instruments, 89 (2018), p. 053704, https://doi.org/10.1063/1.5037341.

[7] A. Beck and M. Teboulle, *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*, SIAM Journal on Imaging Sciences, 2 (2009), pp. 183–202, https://doi.org/10.1137/080716542.

[8] S. Bellavia, *Inexact interior-point method*, Journal of Optimization Theory and Applications, 96 (1998), pp. 109–121, https://doi.org/10.1023/A:1022663100715.

[9] L. Bergamaschi, J. Gondzio, A. Martínez, J. W. Pearson, and S. Pougkakiotis, *A new preconditioning approach for an interior point-proximal method of multipliers for linear and convex quadratic programming*, Numerical Linear Algebra with Applications, p. e2361, https://doi.org/10.1002/nla.2361.

[10] M. Bertero, P. Boccacci, G. Desiderà, and G. Vicidomini, *Image deblurring with Poisson data: from cells to galaxies*, Inverse Problems, 25 (2009), p. 123006, https://doi.org/10.1088/0266-5611/25/12/123006.

[11] D. P. Bertsekas, *Nonlinear programming*, Athena Scientific Optimization and Computation Series, Athena Scientific, Belmont, MA, second ed., 1999.

[12] 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 (2011), pp. 1–122, https://doi.org/10.1561/2200000016.

[13] S. Cafieri, M. D'Apuzzo, V. De Simone, and D. di Serafino, *On the iterative solution of KKT systems in potential reduction software for large-scale quadratic problems*, Computational Optimization and Applications, 38 (2007), pp. 27–45, https://doi.org/10.1007/s10589-007-9035-y.

[14] S. Cafieri, M. D'Apuzzo, V. De Simone, D. di Serafino, and G. Toraldo, *Convergence analysis of an inexact potential reduction method for convex quadratic programming*, Journal of Optimization Theory and Applications, 135 (2007), pp. 355–366, https://doi.org/10.1007/s10957-007-9264-3.

[15] E. J. Candés, J. Romberg, and T. Tao, *Stable signal recovery from incomplete and inaccurate measurements*, Communications in Pure Applied Mathematics, 59 (2006), pp. 1207–1223, https://doi.org/10.1002/cpa.20124.

[16] A. Chambolle, *An algorithm for total variation minimization and applications*, Journal of Mathematical Imaging and Vision, 20 (2004), pp. 89–97, https://doi.org/10.1023/B:JMIV.0000011325.36760.1e.

[17] C. Chen, T. Liang, and G. Biros, *RCHOL: randomized Cholesky factorization for solving SDD linear systems*, 2021, https://arxiv.org/abs/2011.07769.

[18] S. S. Chen, D. L. Donoho, and M. A. Saunders, *Atomic decomposition by basis pursuit*, SIAM Review, 43 (2001), pp. 129–159, https://doi.org/10.1137/S003614450037906X.

[19] Z.-p. Chen, G. Li, and J.-e. Guo, *Optimal investment policy in the time consistent mean-variance formulation*, Insurance: Mathematics & Economics, 52 (2013), pp. 145–156, https://doi.org/10.1016/j.insmatheco.2012.11.007.

[20] A. Chowdhury, P. London, H. Avron, and P. Drineas, *Speeding up linear programming using randomized linear algebra*, arXiv:2003.08072, (2020).

[21] A. Chowdhury, J. Yang, and P. Drineas, *An iterative, sketching-based framework for ridge*

regression, Proceedings of the 35th International Conference on Machine Learning, 80 (2018), pp. 989–998, http://proceedings.mlr.press/v80/chowdhury18a.html.

[22] S. Corsaro and V. De Simone, *Adaptive l_1-regularization for short-selling control in portfolio selection*, Computational Optimization and Applications, 72 (2019), pp. 457–478, https://doi.org/10.1007/s10589-018-0049-4.

[23] S. Corsaro, V. De Simone, and Z. Marino, *Fused lasso approach in portfolio selection*, Annals of Operations Research, 299 (2021), pp. 47–59, https://doi.org/10.1007/s10479-019-03289-w.

[24] S. Corsaro, V. De Simone, and Z. Marino, *Split Bregman iteration for multi-period mean variance portfolio optimization*, Applied Mathematics and Computation, 392 (2021), pp. 125715, 10, https://doi.org/10.1016/j.amc.2020.125715.

[25] S. Corsaro, V. De Simone, Z. Marino, and F. Perla, *l_1-regularization for multi-period portfolio selection*, Annals of Operations Research, 294 (2020), pp. 75–86, https://doi.org/10.1007/s10479-019-03308-w.

[26] Y.-H. Dai and R. Fletcher, *New algorithms for singly linearly constrained quadratic programs subject to lower and upper bounds*, Mathematical Programming, 106 (2006), pp. 403–421, https://doi.org/10.1007/s10107-005-0595-2.

[27] M. D'Apuzzo, V. De Simone, and D. di Serafino, *On mutual impact of numerical linear algebra and large-scale optimization with focus on interior point methods*, Computational Optimization and Applications, 45 (2010), pp. 283–310, https://doi.org/10.1007/s10589-008-9226-1.

[28] V. De Simone, D. di Serafino, and M. Viola, *A subspace-accelerated split Bregman method for sparse data recovery with joint ℓ_1-type regularizers*, Electronic Transactions on Numerical Analysis, 53 (2020), pp. 406–425, https://doi.org/10.1553/etna_vol53s406.

[29] D. di Serafino, G. Landi, and M. Viola, *ACQUIRE: an inexact iteratively reweighted norm approach for TV-based Poisson image restoration*, Applied Mathematics and Computation, 364 (2020), pp. 124678, 23, https://doi.org/10.1016/j.amc.2019.124678.

[30] D. di Serafino and D. Orban, *Constraint-preconditioned Krylov solvers for regularized saddle-point systems*, SIAM Journal on Scientific Computing, 43 (2021), pp. A1001–A1026, https://doi.org/10.1137/19M1291753.

[31] E. D. Dohmatob, A. Gramfort, B. Thirion, and G. Varoquaux, *Benchmarking solvers for TV-ℓ1 least-squares and logistic regression in brain imaging*, in 2014 International Workshop on Pattern Recognition in Neuroimaging, June 2014, pp. 1–4, https://doi.org/10.1109/PRNI.2014.6858516.

[32] M. Dubois, F. Hadj-Selem, T. Löfstedt, M. Perrot, C. Fischer, V. Frouin, and E. Duchesnay, *Predictive support recovery with TV-Elastic Net penalty and logistic regression: An application to structural MRI*, in 2014 International Workshop on Pattern Recognition in Neuroimaging, June 2014, pp. 1–4, https://doi.org/10.1109/PRNI.2014.6858517.

[33] K. Fountoulakis and J. Gondzio, *A second-order method for strongly convex ℓ_1-regularization problems*, Mathematical Programming, 156 (2016), pp. 189–219, https://doi.org/10.1007/s10107-015-0875-4.

[34] K. Fountoulakis, J. Gondzio, and P. Zhlobich, *Matrix-free interior point method for compressed sensing problems*, Mathematical Programming Computation, 6 (2014), pp. 1–31, https://doi.org/10.1007/s12532-013-0063-6.

[35] M. P. Friedlander and D. Orban, *A primal-dual regularized interior-point method for convex quadratic programs*, Mathematical Programming Computation, 4 (2012), pp. 71–107, https://doi.org/10.1007/s12532-012-0035-2.

[36] M. P. Friedlander and P. Tseng, *Exact regularization of convex programs*, SIAM Journal on Optimization, 18 (2007), pp. 1326–1350, https://doi.org/10.1137/060675320.

[37] J. Gondzio, *Interior point methods 25 years later*, European Journal of Operational Research, 218 (2012), pp. 587–601, https://doi.org/10.1016/j.ejor.2011.09.017.

[38] J. Gondzio, *Convergence analysis of an inexact feasible interior point method for convex quadratic programming*, SIAM Journal on Optimization, 23 (2013), pp. 1510–1527, https://doi.org/10.1137/120886017.

[39] J. Gondzio and M. Makowski, *Solving a class of LP problems with a primal-dual logarithmic barrier method*, European Journal of Operational Research, 80 (1995), pp. 184–192, https://doi.org/https://doi.org/10.1016/0377-2217(93)E0323-P.

[40] J. Gondzio and G. Toraldo (eds.), *Linear algebra issues arising in interior point methods*, Special issue of Computational Optimization and Applications, 36 (2007), pp. 137–341.

[41] A. Gramfort, B. Thirion, and G. Varoquaux, *Identifying predictive regions from fMRI with TV-L1 prior*, in 2013 International Workshop on Pattern Recognition in Neuroimaging, June 2013, pp. 17–20, https://doi.org/10.1109/PRNI.2013.14.

[42] L. Grosenick, B. Klingenberg, K. Katovich, B. Knutson, and J. E. Taylor, *Interpretable whole-brain prediction analysis with graphnet*, NeuroImage, 72 (2013), pp. 304–321, https://doi.org/10.1016/j.neuroimage.2012.12.062.

[43] P. C. Hansen, J. G. Nagy, and D. P. O'Leary, *Deblurring Images*, Society for Industrial and Applied Mathematics, 2006, https://doi.org/10.1137/1.9780898718874.

[44] M. R. Hestenes and E. Stiefel, *Methods of conjugate gradients for solving linear systems*, J. Res. Natl. Bur. Stand, 49 (1952), pp. 409–436.

[45] B. Jie, C.-Y. Wee, D. Shen, and D. Zhang, *Hyper-connectivity of functional networks for brain disease diagnosis*, Medical Image Analysis, 32 (2016), pp. 84–100, https://doi.org/10.1016/j.media.2016.03.003.

[46] Y. Kamitani and F. Tong, *Decoding the visual and subjective contents of the human brain*, Nature Neuroscience, 8 (2005), pp. 679–685, https://doi.org/10.1038/nn1444.

[47] D. Klatte and B. Kummer, *Nonsmooth Equations in Optimization, Regularity, calculus, methods and applications*, vol. 60 of Nonconvex Optimization and its Applications, Kluwer Academic Publishers, Dordrecht, Springer, Boston, MA, 2002, https://doi.org/10.1007/b130810.

[48] M. Kojima and S. Shindo, *Extension of Newton and quasi-Newton methods to systems of PC^1 equations*, Journal of the Operational Research Society of Japan, 29 (1986), pp. 352–375, https://doi.org/10.15807/jorsj.29.352.

[49] R. Kyng and S. Sachdeva, *Approximate Gaussian elimination for Laplacians - fast, sparse, and simple*, 2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS), 1 (2016), pp. 573–582, https://doi.org/10.1109/FOCS.2016.68.

[50] J. D. Lee, Y. Sun, and M. A. Saunders, *Proximal Newton-type methods for minimizing composite functions*, SIAM Journal on Optimization, 24 (2014), pp. 1420–1443, https://doi.org/10.1137/130921428.

[51] D. Li and W. Ng, *Optimal dynamic portfolio selection: Multiperiod mean-variance formulation*, Mathematical Finance, 10 (2000), pp. 387–406, https://doi.org/10.1111/1467-9965.00100.

[52] X. Li, D. Sun, and K.-C. Toh, *A highly efficient semismooth Newton augmented Lagrangian method for solving lasso problems*, SIAM Journal on Optimization, 28 (2017), pp. 433–458, https://doi.org/10.1137/16M1097572.

[53] Y. Li, C. Sun, P. Li, Y. Zhao, G. K. Mensah, Y. Xu, H. Guo, and J. Chen, *Hypernetwork construction and feature fusion analysis based on sparse group lasso method on fMRI dataset*, Frontiers in Neuroscience, 14 (2020), https://doi.org/10.3389/fnins.2020.00060.

[54] Y. Malitsky and T. Pock, *A first-order primal-dual algorithm with linesearch*, SIAM Journal on Optimization, 28 (2018), pp. 411–432, https://doi.org/10.1137/16M1092015.

[55] H. M. Markowitz, *Portfolio selection: Efficient diversification of investments*, Cowles Foundation for Research in Economics at Yale University, Monograph 16, John Wiley & Sons, Inc., New York; Chapman & Hall, Ltd., London, 1959.

[56] S. Mehrotra, *On the implementation of a primal-dual interior point method*, SIAM Journal on Optimization, 2 (1992), pp. 575–601, https://doi.org/10.1137/0802028.

[57] V. Michel, A. Gramfort, G. Varoquaux, E. Eger, and B. Thirion, *Total variation regularization for fMRI-based prediction of behavior*, IEEE Transactions on Medical Imaging, 30 (2011), pp. 1328–1340, https://doi.org/10.1109/TMI.2011.2113378.

[58] A. M. Mota, N. Oliveira, P. Almeida, and N. Matela, *3D total variation minimization filter for breast tomosynthesis imaging*, in Breast Imaging, A. Tingberg, K. Lång, and P. Timberg, eds., Cham, 2016, Springer, pp. 501–509, https://doi.org/10.1007/978-3-319-41546-8_63.

[59] J. Mourão-Miranda, E. Reynaud, F. McGlone, G. Calvert, and M. Brammer, *The impact of temporal compression and space selection on SVM analysis of single-subject and multi-subject fMRI data*, NeuroImage, 33 (2006), pp. 1055–1065, https://doi.org/10.1016/j.neuroimage.2006.08.016.

[60] Y. Nesterov and A. Nemirovskii, *Interior-point polynomial algorithms in convex programming*, vol. 13 of SIAM Studies in Applied Mathematics, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1994, https://doi.org/10.1137/1.9781611970791.

[61] C. C. Paige and M. A. Saunders, *Solution of sparse indefinite systems of linear equations*, SIAM Journal on Numerical Analysis, 12 (1975), pp. 617–629, https://doi.org/10.1137/0712047.

[62] R. Peng and D. A. Spielman, *An efficient parallel solver for sdd linear systems*, in Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing, New York, NY, USA, 2014, Association for Computing Machinery, p. 333–342, https://doi.org/10.1145/2591796.2591832.

[63] S. Pougkakiotis and J. Gondzio, *Dynamic non-diagonal regularization in interior point methods for linear and convex quadratic programming*, Journal of Optimization Theory and Applications, 181 (2019), pp. 905–945, https://doi.org/10.1007/s10957-019-01491-1.

[64] S. Pougkakiotis and J. Gondzio, *An interior point-proximal method of multipliers for convex quadratic programming*, Computational Optimization and Applications, 78 (2021), pp. 307–351, https://doi.org/10.1007/s10589-020-00240-9.

[65] S. Pougkakiotis and J. Gondzio, *An interior point-proximal method of multipliers for linear positive semi-definite programming*, Journal of Optimization Theory and Applications, (2021), https://doi.org/10.1007/s10957-021-01954-4.

[66] M. J. Rosa, L. Portugal, T. Hahn, A. J. Fallgatter, M. I. Garrido, J. Shawe-Taylor, and J. Mourão-Miranda, *Sparse network-based models for patient classification using fMRI*, NeuroImage, 105 (2015), pp. 493–506, https://doi.org/10.1016/j.neuroimage.2014.11.021.

[67] L. I. Rudin, S. Osher, and E. Fatemi, *Nonlinear total variation based noise removal algorithms*, Physica D, 60 (1992), pp. 259–268, https://doi.org/10.1016/0167-2789(92)90242-F.

[68] S. Ryali, K. Supekar, D. A. Abrams, and V. Menon, *Sparse logistic regression for whole-brain classification of fMRI data*, NeuroImage, 51 (2010), pp. 752–764, https://doi.org/10.1016/j.neuroimage.2010.02.040.

[69] M. Saunders and J. A. Tomlin, *Solving regularized linear programs using barrier methods and KKT systems*, Tech. Report SOL 96-4, Systems Optimization Laboratory, Department of Operations Research, Stanford University, Stanford, CA 94305, USA, December 1996.

[70] M. A. Saunders, *Cholesky-based methods for sparse least squares: the benefits of regularization*, in Linear and nonlinear conjugate gradient-related methods (Seattle, WA, 1995), SIAM, Philadelphia, PA, 1996, pp. 92–100.

[71] M. Schmidt, D. Kim, and S. Sra, *Projected Newton-type methods in machine learning*, Optimization for Machine Learning, MIT Press, 2011, pp. 305–330, https://doi.org/10.7551/mitpress/8996.003.0013.

[72] R. Tapia, Y. Zhang, M. Saltzman, and A. Weiser, *The Mehrotra predictor-corrector interior-point method as a perturbed composite Newton method*, SIAM Journal on Optimization, 6 (1996), pp. 47–56, https://doi.org/10.1137/0806004.

[73] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight, *Sparsity and smoothness via the fused lasso*, Journal of the Royal Statistical Society: Series B (Statistical Methodology), 67 (2005), pp. 91–108, https://doi.org/10.1111/j.1467-9868.2005.00490.x.

[74] J. A. Tropp and S. J. Wright, *Computational methods for sparse solution of linear inverse problems*, Proceedings of the IEEE, 98 (2010), pp. 948–958, https://doi.org/10.1109/JPROC.2010.2044010.

[75] R. J. Vanderbei, *Symmetric quasidefinite matrices*, SIAM Journal on Optimization, 5 (1995), pp. 100–113, https://doi.org/10.1137/0805005.

[76] V. N. Vapnik, *Statistical learning theory*, John Wiley & Sons, New York, 1998.

[77] R. A. Waltz, J. L. Morales, J. Nocedal, and D. Orban, *An interior algorithm for nonlinear optimization that combines line search and trust region steps*, Mathematical Programming, 107 (2006), pp. 391–408, https://doi.org/10.1007/s10107-004-0560-5.

[78] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, *Image quality assessment: from error visibility to structural similarity*, IEEE Transactions on Image Processing, 13 (2004), pp. 600–612, https://doi.org/10.1109/TIP.2003.819861.

[79] Y.-W. Wen, R. H. Chan, and T.-Y. Zeng, *Primal-dual algorithms for total variation based image restoration under Poisson noise*, Science China Mathematics, 59 (2016), pp. 141–160, https://doi.org/10.1007/s11425-015-5079-0.

[80] O. Yamashita, M. Sato, T. Yoshioka, F. Tong, and Y. Kamitani, *Sparse estimation automatically selects voxels relevant for the decoding of fMRI activity patterns*, NeuroImage, 42 (2008), pp. 1414–1429, https://doi.org/10.1016/j.neuroimage.2008.05.050.

[81] G.-X. Yuan, C.-H. Ho, and C.-J. Lin, *An improved GLMNET for L1-regularized logistic regression*, Journal of Machine Learning Research, 13 (2012), pp. 1999–2030, http://jmlr.org/papers/v13/yuan12a.html.

[82] M.-C. Yue, Z. Zhou, and A. M.-C. So, *A family of inexact SQA methods for non-smooth convex minimization with provable convergence guarantees based on the Luo-Tseng error bound property*, Mathematical Programming, 174 (2019), pp. 327–358, https://doi.org/10.1007/s10107-018-1280-6.

[83] J. Zhang, Y. Hu, and J. G. Nagy, *A scaled gradient method for digital tomographic image reconstruction*, Inverse Problems & Imaging, 12 (2018), pp. 239–259, https://doi.org/10.3934/ipi.2018010.

[84] Y. Zhang, *On the convergence of a class of infeasible interior-point methods for the horizontal*

linear complementarity problem, SIAM Journal on Optimization, 4 (1994), pp. 208–227, https://doi.org/10.1137/0804012.