The Fastest $\ell_{1,\infty}$ Prox in the West

Benjamín Béjar, Ivan Dokmanić, *Member, IEEE*, and René Vidal, *Fellow, IEEE*

Abstract—Proximal operators are of particular interest in optimization problems dealing with non-smooth objectives because in many practical cases they lead to optimization algorithms whose updates can be computed in closed form or very efficiently. A well-known example is the proximal operator of the vector ℓ_1 norm, which is given by the soft-thresholding operator. In this paper we study the proximal operator of the mixed $\ell_{1,\infty}$ matrix norm and show that it can be computed in closed form by applying the well-known soft-thresholding operator to each column of the matrix. However, unlike the vector ℓ_1 norm case where the threshold is constant, in the mixed $\ell_{1,\infty}$ norm case each column of the matrix might require a different threshold and all thresholds depend on the given matrix. We propose a general iterative algorithm for computing these thresholds, as well as two efficient implementations that further exploit easy to compute lower bounds for the mixed norm of the optimal solution. Experiments on large-scale synthetic and real data indicate that the proposed methods can be orders of magnitude faster than state-of-the-art methods.

Index Terms—Proximal operator, mixed norm, block sparsity

1 Introduction

Recent advances in machine learning and convex optimization techniques have led to very efficient algorithms for solving a family of regularized estimation problems. Sparsity, as a strong regularization prior, plays a central role in many inverse problems and the use of sparsity-promoting norms as regularizers has become widespread over many different disciplines of science and engineering. One added difficulty is the non-differentiability of such priors, which prevents the use of classical optimization methods such as gradient descent or Gauss-Newton methods [1], [2]. Proximal algorithms present an efficient alternative to cope with non-smoothness of the objective function. Furthermore, in many practical situations, simple closed-form updates of the variables of interest are possible. For an excellent review about proximal operators and algorithms see [3] and the monographs [4], [5].

1.1 Motivation

Let $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m] \in \mathbb{R}^{n \times m}$ be a real matrix with columns $\boldsymbol{x}_i \in \mathbb{R}^n$. The mixed $\ell_{p,q}$ norm of $\boldsymbol{X}$ is defined over its columns as

$$\|\boldsymbol{X}\|_{p,q} = \left(\sum_{i=1}^{m} \|\boldsymbol{x}_i\|_p^q \right)^{1/q}. \tag{1}$$

Mixed norms such as the $\ell_{p,1}$ matrix norm ($p \geq 2$) have been used to promote block-sparse structure in the variables of interest, and the larger p the stronger the correlation among the rows of $\boldsymbol{X}$ [6]. In particular, the $\ell_{\infty,1}$ norm has been shown to be useful for estimating a set of covariate regressors in problems such as multi-task learning [6], [7], [8], and representative (exemplar) selection [9]. A general formulation for these type of problems is to minimize some convex loss function subject to norm constraints:

$$\begin{aligned} \underset{\boldsymbol{X}}{\text{minimize}} \quad & J(\boldsymbol{X}) \\ \text{subject to} \quad & \|\boldsymbol{X}\|_{\infty,1} \leq \tau, \end{aligned} \tag{2}$$

where $\tau > 0$ controls the sparsity level and $J(\cdot)$ is some convex loss function. Note that keeping $\|\boldsymbol{X}\|_{\infty,1}$ small encourages whole columns of $\boldsymbol{X}$ to be zero. In this contribution, we are interested in efficiently solving problems of the form of (2). A simple method to solve problem (2) is to use a projected (sub)gradient descent method that computes the kth iteration estimate $\boldsymbol{X}^{(k)}$ as

$$\boldsymbol{Z} \leftarrow \boldsymbol{X}^{(k-1)} - \eta_k \partial J\big(\boldsymbol{X}^{(k-1)}\big) \tag{3}$$

$$\boldsymbol{X}^{(k)} \leftarrow \mathrm{P}_{\|\cdot\|_{\infty,1} \leq \tau}(\boldsymbol{Z}), \tag{4}$$

where η_k is the stepsize at the kth iteration of the algorithm, $\partial J(\boldsymbol{X})$ denotes a subgradient (i.e., the gradient if differentiable) of J at $\boldsymbol{X}$, and where $\mathrm{P}_{\|\cdot\|_{\infty,1} \leq \tau}(\boldsymbol{Z})$ denotes the projection of $\boldsymbol{Z}$ onto the $\ell_{\infty,1}$ ball of radius τ. In solving problems of the form of (2) one needs to compute a projection onto the $\ell_{\infty,1}$ mixed norm ball. Such projection can be computed by a proximal mapping of the dual norm – the mixed $\ell_{1,\infty}$ norm (i.e., the induced ℓ_1 norm of $\boldsymbol{X}$ seen as a linear operator). In this paper, we address the problem of projecting onto the mixed $\ell_{\infty,1}$ norm ball via the computation of the proximal operator of its dual norm. This allows us to solve the class of problems in (2) that involve structured/group sparsity, namely those involving constraints on (projections onto) the mixed $\ell_{\infty,1}$ norm. The proximal mapping of the mixed $\ell_{1,\infty}$ norm is also applicable to the computation of minimax sparse pseudoinverses to underdetermined systems of linear equations [10], [11].

- *Benjamín Béjar and René Vidal are with the Mathematical Institute for Data Science, Department of Biomedical Engineering, Johns Hopkins University, Baltimore, MD 21218 USA. E-mail: {bbejar, rvidal}@jhu.edu.*
- *Ivan Dokmanić is with the Department of Electrical and Computer Engineering, University of Illinois at Urbana Champaign, Champaign, IL 61820 USA. E-mail: dokmanic@illinois.edu.*

Manuscript received 12 July 2019; revised 23 Nov. 2020; accepted 30 Jan. 2021. Date of publication 15 Feb. 2021; date of current version 3 June 2022. (Corresponding author: Benjamín Béjar.)
Recommended for acceptance by O. Camps.
Digital Object Identifier no. 10.1109/TPAMI.2021.3059301

1.2 Prior Work

Since one of the computational challenges in solving problems of the form (2) is in the computation of the projection onto the $\ell_{\infty,1}$ ball of a certain radius, it is then of practical importance to devise computationally efficient algorithms for computing such projections. An efficient method for computing these projections is proposed in [8]. The algorithm is based on sorting the entries of the n by m data matrix in order to find the values that satisfy the optimality conditions of the projection problem. The complexity of the method is dominated by the sorting operation and therefore has an average complexity of $O(mn\log(mn))$. An alternative strategy is to use root-search methods such as those in [12], [13] in order to find the optimal solution. Here we take an alternative approach and look at the proximal operator of the mixed $\ell_{1,\infty}$ matrix norm. Since the mixed $\ell_{1,\infty}$ and $\ell_{\infty,1}$ norms are duals of each other, a simple relationship can be established between the proximal operator and the projection operator (see Section 2). However, by looking at the proximal operator a better insight and understanding of the problem can be gained and exploited to accelerate the algorithms. Contrary to root-search methods our method is exact (up to machine precision), does not require any thresholds to determine convergence, and it is guaranteed to find the optimal solution in a finite number of iterations.

1.3 Contributions

In this paper we study the proximal operator of the mixed $\ell_{1,\infty}$ matrix norm and show that it can be computed using a generalization of the well-known soft-thresholding operator from the vector to the matrix case. The generalization involves applying the soft-thresholding operator to each column of the matrix using a possibly different threshold for each column. Interestingly, all thresholds are related to each other via a quantity that depends on the given matrix. This is in sharp contrast to the vector case, where the threshold is constant and is given by the regularization parameter. To compute the proximal operator efficiently, we propose a general iterative algorithm based on the optimality conditions of the proximal problem. Our method is further accelerated by the derivation of easy to compute lower bounds on the optimal value of the proximal problem that contribute to effectively reduce the search space. A numerical comparison with the state of the art of two particular implementations of our general method reveals the improved computational efficiency of the proposed algorithms. We also illustrate the application of our results to biomarker discovery for the problem of cancer classification from gene expression data. The code used to generate the results presented in this paper is made publicly available by the authors.[1]

2 Norms, Projections, and Proximal Operators

In this section we present some background material that highlights the relationship between proximal operators, norms, and orthogonal projection operators.

Consider a non-empty closed convex set $\mathcal{C} \subset \mathbb{R}^n$. The orthogonal projection of a point $\boldsymbol{x} \in \mathbb{R}^n$ onto $\mathcal{C}$ is given by

$$\mathrm{P}_{\mathcal{C}}(\boldsymbol{x}) = \operatorname*{argmin}_{\boldsymbol{y}\in\mathcal{C}} \frac{1}{2}\|\boldsymbol{x}-\boldsymbol{y}\|_2^2, \tag{5}$$

where we have included an irrelevant $1/2$ factor for convenience in the exposition. Alternatively, we can also express the projection of a point as an unconstrained optimization problem as

$$\mathrm{P}_{\mathcal{C}}(\boldsymbol{x}) = \operatorname*{argmin}_{\boldsymbol{y}} \mathbb{I}(\boldsymbol{y}\in\mathcal{C}) + \frac{1}{2}\|\boldsymbol{x}-\boldsymbol{y}\|_2^2, \tag{6}$$

where we have moved the constraint into the objective by making use of the indicator function of a non-empty subset $\mathcal{X} \subset \mathbb{R}^n$, which is given by

$$\mathbb{I}(\boldsymbol{x}\in\mathcal{X}) = \begin{cases} 0, & \boldsymbol{x}\in\mathcal{X} \\ +\infty, & \text{otherwise} \end{cases}. \tag{7}$$

Keeping in mind the definition of the projection operator given in (6) as an unconstrained optimization problem, we are now ready to introduce the definition of the proximal operator. Let $f(\boldsymbol{x}) : \mathbb{R}^n \mapsto \mathbb{R}$ be a lower semicontinuous convex function. Then, for every $\boldsymbol{x} \in \mathbb{R}^n$ the proximal operator $\mathrm{prox}_f(x)$ is defined as

$$\mathrm{prox}_f(\boldsymbol{x}) = \operatorname*{argmin}_{\boldsymbol{y}} f(\boldsymbol{y}) + \frac{1}{2}\|\boldsymbol{x}-\boldsymbol{y}\|_2^2. \tag{8}$$

It is then clear, that the proximal operator can be regarded as a generalization of the projection operator (e.g., replace $f(\boldsymbol{y})$ by the indicator function of a set $\mathcal{C}$). Note that, at every point, the proximal operator is the *unique* solution of an unconstrained convex optimization problem. Uniqueness of the proximal operator can be easily argued from the fact that the quadratic term in (8) makes the optimization cost strictly convex.

An important particular case that often appears in practice is that where the function f is a norm. For example, problems of the form of (8) appear in many learning and signal processing problems, where the quadratic term can be seen as a data-fidelity term while the function f can be thought of as imposing some prior on the solution (e.g., sparsity). The special case where f is a norm has also a close connection to projections via the Moreau decomposition theorem as we shall describe next. Let $f : \mathcal{X} \subseteq \mathbb{R}^n \mapsto \mathbb{R}$ be a lower semicontinuous convex function, then its Fenchel conjugate f^* is defined as

$$f^*(\boldsymbol{y}) = \sup_{\boldsymbol{x}\in\mathcal{X}} \big\{\langle \boldsymbol{y}, \boldsymbol{x}\rangle - f(\boldsymbol{x})\big\}. \tag{9}$$

The Moreau decomposition theorem relates the proximal operators of a convex function and its Fenchel conjugate, as stated next.

Theorem 1 ([14]). *Let f be a lower semicontinuous convex function and let f^* denote its Fenchel (or convex) conjugate, then*

$$\mathrm{prox}_f(\boldsymbol{x}) + \mathrm{prox}_{f^*}(\boldsymbol{x}) = \boldsymbol{x}. \tag{10}$$

For the special case where $f(\boldsymbol{x}) = \|\boldsymbol{x}\|$ is a norm, it is well known that its Fenchel conjugate f^* is given by

$$f^*(\boldsymbol{x}) = \mathbb{I}(\|\boldsymbol{x}\|_* \le 1) = \begin{cases} 0, & \|\boldsymbol{x}\|_* \le 1 \\ +\infty, & \text{otherwise} \end{cases}, \tag{11}$$

1. https://github.com/bbejar/prox-l1oo.

where $\|\cdot\|_*$ is the dual norm of $\|\cdot\|$ (i.e., $\|\boldsymbol{z}\|_* = \sup_{\boldsymbol{x}} \{\langle \boldsymbol{z}, \boldsymbol{x}\rangle : \|\boldsymbol{x}\| \leq 1\}$). That is, the Fenchel conjugate of a norm is the indicator function of the unit-norm ball of its dual norm (see for instance [2] for a proof). Since the proximal operator of the indicator function of a set equals the orthogonal projection onto the set, it follows from (10) that

$$\operatorname{prox}_{\lambda\|\cdot\|} = I - \mathrm{P}_{\|\cdot\|_* \leq \lambda}, \tag{12}$$

where $\mathrm{P}_{\|\cdot\|_* \leq \lambda}$ denotes the projection onto the ball of radius λ of the dual norm, and where I is the identity operator.

For a given matrix $\boldsymbol{X} \in \mathbb{R}^{n\times m}$ its mixed $\ell_{1,\infty}$ (induced ℓ_1) norm is given by

$$\|\boldsymbol{X}\|_{1,\infty} = \max_{\|\boldsymbol{u}\|_1 = 1} \|\boldsymbol{X}\boldsymbol{u}\|_1 = \max_i \|\boldsymbol{x}_i\|_1, \tag{13}$$

where $\boldsymbol{x}_i$ corresponds to the ith column of matrix $\boldsymbol{X}$. For the case of the induced ℓ_∞ operator norm we have the well-known relationship

$$\|\boldsymbol{X}\|_\infty = \max_{\|\boldsymbol{u}\|_\infty = 1} \|\boldsymbol{X}\boldsymbol{u}\|_\infty = \|\boldsymbol{X}^\mathsf{T}\|_{1,\infty}. \tag{14}$$

Also, recall the duality relationship between the $\ell_{\infty,1}$ norm and the mixed $\ell_{1,\infty}$ norm:

$$\|\boldsymbol{X}\|_{\infty,1} = \sum_{i=1}^{m} \|\boldsymbol{x}_i\|_\infty = \big(\|\boldsymbol{X}\|_{1,\infty}\big)_*. \tag{15}$$

Thus, without loss of generality, we will focus our analysis on the derivation of the proximal operator for the mixed $\ell_{1,\infty}$ norm, and derive expressions for the proximal operators of the induced ℓ_∞ and the projection operator onto the $\ell_{\infty,1}$ norm using the above relationships.

3 Analysis of the Mixed $\ell_{1,\infty}$ Norm Proximal Operator

The relationship given in (12) makes it clear that finding the proximal operator of a norm amounts to knowing how to project onto the unit-norm ball of the dual norm and vice-versa. In [8] the authors derived the optimality conditions for the projection onto the $\ell_{\infty,1}$ norm (see (15)) and proposed an algorithm for its computation based on sorting the entries of the matrix. Since these norms are duals of each other, the proximal operator for such norms can be readily computed based on (12). In contrast, we look at the proximal operator itself and derive the optimality conditions. By doing so, we arrive at a more compact expression for the optimality conditions that generalizes the well-known soft-thresholding algorithm to the matrix case. Our analysis allows for a more intuitive interpretation of the proximal operator as well as the derivation of novel algorithms for its computation.

Given a matrix $\boldsymbol{V} \in \mathbb{R}^{n\times m}$, the proximal operator of the mixed $\ell_{1,\infty}$ norm with parameter $\lambda > 0$ is the solution to the following convex optimization problem:

$$\operatorname{prox}_{\lambda\|\cdot\|_{1,\infty}}(\boldsymbol{V}) = \operatorname*{argmin}_{\boldsymbol{X}} \|\boldsymbol{X}\|_{1,\infty} + \frac{1}{2\lambda}\|\boldsymbol{X} - \boldsymbol{V}\|_F^2. \tag{16}$$

Using the definition of the mixed norm in (13), we can rewrite problem (16) as the following constrained optimization problem:

$$\begin{aligned} \underset{\boldsymbol{X},t}{\text{minimize}} \quad & t + \frac{1}{2\lambda}\|\boldsymbol{X} - \boldsymbol{V}\|_F^2 \\ \text{subject to} \quad & \|\boldsymbol{x}_i\|_1 \leq t, \quad i = 1,\ldots,m. \end{aligned} \tag{17}$$

By looking at the structure of problem (17) it is easy to derive the following result:

Lemma 1 (Matched Sign). *The sign of the optimal solution $\boldsymbol{X}^\star$ of (17) must match the sign of $\boldsymbol{V}$, that is*

$$\operatorname{sign}(\boldsymbol{X}^\star) = \operatorname{sign}(\boldsymbol{V}), \tag{18}$$

where the $\operatorname{sign}(\cdot)$ *function operates element-wise.*

Proof. The proof follows by contradiction. Assume $(\boldsymbol{X}^\star, t^\star)$ is the optimal solution to problem (17) and that there are some nonzero entries of $\boldsymbol{X}^\star$ that have the opposite sign to the corresponding entries in $\boldsymbol{V}$, i.e., $\operatorname{sign}(x^\star_{ij}) = -\operatorname{sign}(v_{ij})$ for some ij. Now, form the matrix $\tilde{\boldsymbol{X}}$ such that $\tilde{x}_{ij} = \operatorname{sign}(v_{ij})|x^\star_{ij}|$. The point $(\tilde{\boldsymbol{X}}, t^\star)$ is feasible and causes a reduction in the objective function since $\|\tilde{\boldsymbol{X}} - \boldsymbol{V}\|_F < \|\boldsymbol{X}^\star - \boldsymbol{V}\|_F$ while keeping the norm unchanged $\|\boldsymbol{X}^\star\|_{1,\infty} = t^\star = \|\tilde{\boldsymbol{X}}\|_{1,\infty}$. This contradicts the assumption that $\boldsymbol{X}^\star$ is the optimal solution. □

Based on Lemma 1 the problem of finding the proximal operator in (17) boils down to finding the magnitudes of the entries of the matrix $\boldsymbol{X}$. Therefore, we can formulate it as[2]

$$\begin{aligned} \underset{\boldsymbol{X},t}{\text{minimize}} \quad & t + \frac{1}{2\lambda}\|\boldsymbol{X} - \boldsymbol{U}\|_F^2 \\ \text{subject to} \quad & \mathbf{1}^\mathsf{T}\boldsymbol{x}_i \leq t, \quad \boldsymbol{x}_i \geq \mathbf{0}, \quad i = 1,\ldots,m, \end{aligned} \tag{19}$$

where $\boldsymbol{U} = [\boldsymbol{u}_1,\ldots,\boldsymbol{u}_m] \in \mathbb{R}^{n\times m}_+$ is a matrix with non-negative entries given by $u_{ij} = |v_{ij}|$. The following result determines the optimal solution of problem (19) and, as a consequence, it also determines the proximal operator of the mixed $\ell_{1,\infty}$ norm:

Proposition 1. *The optimal solution $(\boldsymbol{X}^\star, t^\star)$ of problem (19) is given by*

$$\boldsymbol{X}^\star = \Big[\boldsymbol{U} - \lambda\mathbf{1}\boldsymbol{\mu}^\mathsf{T}\Big]_+, \tag{20}$$

and

$$t^\star = \frac{\sum_{i\in\mathcal{M}^\star} \frac{1}{|\mathcal{J}_i^\star|}\sum_{j\in\mathcal{J}_i^\star} u_{ij} - \lambda}{\sum_{i\in\mathcal{M}^\star}\frac{1}{|\mathcal{J}_i^\star|}}, \tag{21}$$

where $[\cdot]_+ = \max(\cdot, 0)$, $\mathcal{M}^\star = \{1 \leq i \leq m : \mathbf{1}^\mathsf{T}\boldsymbol{u}_i \geq t^\star\}$ *is the set of columns affected by thresholding,* $\mathcal{J}_i^\star = \{1 \leq j \leq n : u_{ij} - \lambda\mu_i^\star \geq 0\}$ *is the set of indices of the non-zero entries of* $\boldsymbol{x}_i^\star$*, and*

$$\mu_i^\star = \Big[\frac{\sum_{j\in\mathcal{J}_i^\star} u_{ij} - t^\star}{\lambda|\mathcal{J}_i^\star|}\Big]_+, \; i = 1,\ldots,m, \tag{22}$$

is the ith entry of the vector $\boldsymbol{\mu} \in \mathbb{R}^m_+$.

2. Notice that this is a power allocation problem which belongs to the general family of waterfilling problems [15].

Proof. The Lagrangian of problem (19) is given by

$$\mathcal{L}\big(\boldsymbol{X}, t, \boldsymbol{\mu}, \{\boldsymbol{\sigma}_i\}_{i=1}^m\big) = t + \frac{1}{2\lambda}\sum_{i=1}^m \|\boldsymbol{x}_i - \boldsymbol{u}_i\|^2 + \sum_{i=1}^m \mu_i(\mathbf{1}^\mathsf{T}\boldsymbol{x}_i - t) - \sum_{i=1}^m \boldsymbol{\sigma}_i^\mathsf{T}\boldsymbol{x}_i \ . \tag{23}$$

Since the problem is convex, the necessary and sufficient conditions for optimality are given by the KKT conditions:

- Zero gradient of the Lagrangian

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{x}_k} = \frac{1}{\lambda}(\boldsymbol{x}_k - \boldsymbol{u}_k) + \mu_k \mathbf{1} - \boldsymbol{\sigma}_k = \mathbf{0}, \ \forall k \tag{24}$$

$$\frac{\partial \mathcal{L}}{\partial t} = 1 - \sum_{i=1}^m \mu_i = 0 \tag{25}$$

- Primal and dual feasibility

$$\mathbf{1}^\mathsf{T}\boldsymbol{x}_k \le t, \quad \boldsymbol{x}_k \ge \mathbf{0}, \quad k = 1, \dots, m \tag{26}$$

$$\boldsymbol{\mu} \ge \mathbf{0}, \quad \boldsymbol{\sigma}_k \ge \mathbf{0}, \quad k = 1, \dots, m \tag{27}$$

- Complementary slackness

$$\mu_k(\mathbf{1}^\mathsf{T}\boldsymbol{x}_k - t) = 0, \quad k = 1, \dots, m \tag{28}$$

$$\boldsymbol{\sigma}_k \odot \boldsymbol{x}_k = 0, \quad k = 1, \dots, m, \tag{29}$$

where $\odot$ denotes element-wise product.

We start by showing that equation (20) holds or equivalently, that every column $\boldsymbol{x}_k$ of $\boldsymbol{X}$ satisfies

$$\boldsymbol{x}_k = \big[\boldsymbol{u}_k - \lambda\mu_k \mathbf{1}\big]_+, \ k = 1, \dots, m. \tag{30}$$

In order to do so, let $\mathcal{M} = \{1 \le i \le m : \mathbf{1}^\mathsf{T}\boldsymbol{u}_i \ge t\}$ be the set of columns that are affected by thresholding. Take for instance $\boldsymbol{x}_k$ for some $k \in \mathcal{M}$, then we have

$$\boldsymbol{x}_k = \boldsymbol{u}_k - \lambda\mu_k\mathbf{1} + \lambda\boldsymbol{\sigma}_k. \tag{31}$$

In this case we can have $x_{kj} > 0$ which, by (29), (26), (27) implies $\sigma_{kj} = 0$. Alternatively, we can have $x_{kj} = 0$ which means $u_{kj} - \lambda\mu_k < 0$. Therefore, both situations can be written in compact form as

$$\boldsymbol{x}_k = \big[\boldsymbol{u}_k - \lambda\mu_k\mathbf{1}\big]_+, \ k \in \mathcal{M}, \tag{32}$$

where the thresholding operation $[\cdot]_+$ is applied element-wise. Alternatively, take $\boldsymbol{x}_k$ for some $k \notin \mathcal{M}$ then from (28) it follows that $\mu_k = 0$ and hence, $\boldsymbol{x}_k = \boldsymbol{u}_k + \lambda\boldsymbol{\sigma}_k$. From (29) and the fact that $\boldsymbol{u}_k \ge \mathbf{0}$ it follows that $\boldsymbol{\sigma}_k = \mathbf{0}$ for all $k \notin \mathcal{M}$. Therefore, we have that

$$\boldsymbol{x}_k = \boldsymbol{u}_k, \ k \notin \mathcal{M}. \tag{33}$$

Since $\mu_k = 0$ for $k \notin \mathcal{M}$ we can put together (32) and (33) into a single expression as in (30). It remains now to derive an expression that relates t and $\{\mu_k\}_{k=1}^m$. We know from (28) and the fact that $\mu_k \neq 0$ for $k \in \mathcal{M}$ that

$$\mathbf{1}^\mathsf{T}\boldsymbol{x}_k = \sum_{j=1}^n \big[u_{kj} - \lambda\mu_k\big]_+ = \sum_{j\in\mathcal{J}_k}(u_{kj} - \lambda\mu_k) = t, \ k \in \mathcal{M}, \tag{34}$$

where we the set $\mathcal{J}_k$ denotes the non-zero entries of $\boldsymbol{x}_k$. Solving for μ_k in (34) leads to

$$\mu_k = \frac{\sum_{j\in\mathcal{J}_k} u_{kj} - t}{\lambda|\mathcal{J}_k|}, \ k \in \mathcal{M}. \tag{35}$$

Recall that for $k \notin \mathcal{M}$ we have $\mu_k = 0$ and $\mathbf{1}^\mathsf{T}\boldsymbol{u}_k < t$ therefore, we can compactly express μ_k as

$$\mu_k = \Big[\frac{\sum_{j\in\mathcal{J}_k} u_{kj} - t^\star}{\lambda|\mathcal{J}_k|}\Big]_+, \ k = 1, \dots, m,$$

and we recover the expression in (22). Finally, using equation (25) it is easy to check that

$$t^\star = \frac{\sum_{k\in\mathcal{M}} \frac{1}{|\mathcal{J}_k|}\sum_{j\in\mathcal{J}_k} u_{kj} - \lambda}{\sum_{k\in\mathcal{M}} \frac{1}{|\mathcal{J}_k|}},$$

which completes the proof. □

We are now ready to derive an expression for the proximal operator of the mixed $\ell_{1,\infty}$ norm as:

Corollary 1 (Proximal Operator). *The proximal operator in (16) is given by*

$$\operatorname{prox}_{\lambda\|\cdot\|_{1,\infty}}(\boldsymbol{V}) = \operatorname{sign}(\boldsymbol{V}) \odot \Big[|\boldsymbol{V}| - \lambda\mathbf{1}\boldsymbol{\mu}^\mathsf{T}\Big]_+, \tag{36}$$

where $\boldsymbol{\mu}$ is given as in Proposition 1.

Proof. It follows directly from Lemma 1 and Proposition 1. □

The expression in Corollary 1 resembles very much the well-known soft-thresholding operator. In fact, the proximal operator of the mixed $\ell_{1,\infty}$ norm applies a soft-thresholding operation to every column of the matrix but with a different threshold value $\lambda\mu_i$ for each column $i = 1, \dots, m$ (see Fig. 1). As expected, the above expression reduces to soft-thresholding for $m = 1$:

Corollary 2 (Soft-thresholding). *In the case $m = 1$ so that $\boldsymbol{V} = \boldsymbol{v} \in \mathbb{R}^n$ is a vector, the proximal operator is given by the well-known soft-thresholding*

$$\operatorname{prox}_{\lambda\|\cdot\|_1}(\boldsymbol{v}) = \operatorname{sign}(\boldsymbol{v}) \odot [|\boldsymbol{v}| - \lambda\mathbf{1}]_+. \tag{37}$$

Proof. By setting $m = 1$ we get from Proposition 1 that $t = \sum_{j\in\mathcal{J}} |v_j| - |\mathcal{J}|\lambda$, where $\mathcal{J}$ is the set of non-zero entries of the optimal vector $\boldsymbol{x}^\star$. Substituting this value into (22) we get that $\mu^\star = 1$. The result then follows from Corollary 1. □

Corollary 3 (Projection onto the $\ell_{\infty,1}$ ball). *The projection onto the $\ell_{\infty,1}$ ball of radius λ is*

$$\mathrm{P}_{\|\cdot\|_{\infty,1}\le\lambda}(\boldsymbol{V}) = \operatorname{sign}(\boldsymbol{V}) \odot \min\big(|\boldsymbol{V}|, \lambda\mathbf{1}\boldsymbol{\mu}^\mathsf{T}\big), \tag{38}$$

with $\boldsymbol{\mu}$ given as in Proposition 1.

Proof. The result follows from Corollary 1, (12) and (15). □

Remark. Note that the results presented in this section can be trivially extended to the case of complex-valued

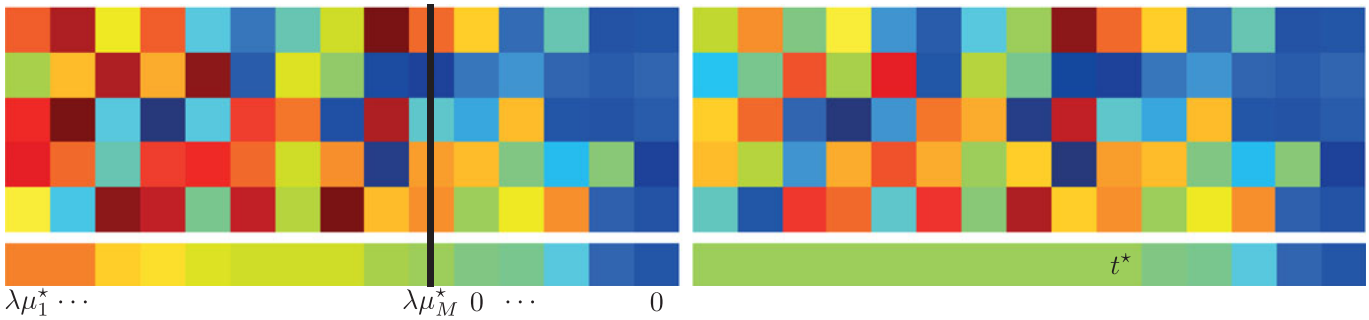

Fig. 1. Illustration of the effect of the proximal operator. The left plot corresponds to the original matrix while the right plot corresponds to its thresholded counterpart. The bottom color plots represent the ℓ_1 norm of each column. Warmer colors mean larger entries. The proximal operator projects the columns of the input matrix onto the ℓ_1 ball of radius $t^\star$.

matrices by interpreting the sign operation as extracting the phase of a complex number (i.e., $\operatorname{sign}(\boldsymbol{V}) = \boldsymbol{V}/|\boldsymbol{V}|$).

4 Algorithms for Computing the Mixed $\ell_{1,\infty}$ Norm Proximal Operator

The results in Proposition 1 and Corollary 1 give us the basis for finding an efficient algorithm for computing the mixed $\ell_{1,\infty}$ norm proximal operator. However, the computation of the proximal operator directly from those expressions requires knowledge about the optimal sets $\mathcal{M}^\star$ and $\{\mathcal{J}_i^\star\}_{i=1}^m$, which are not known a priori. In this section we present a procedure for addressing this issue. But first, we describe an efficient pre-processing stage that can be used to reduce the search space for the optimal sets $\mathcal{M}^\star$ and $\{\mathcal{J}_i^\star\}_{i=1}^m$ needed to compute the proximal operator in Proposition 1. The idea is to maximize a lower bound on the mixed $\ell_{1,\infty}$ norm of the optimal solution, which allows us to discard columns that will not be affected by thresholding hence, reducing the search space for the optimal sets $\mathcal{M}^\star$ and $\{\mathcal{J}_i^\star\}_{i=1}^m$. This allows us to effectively reduce the dimensionality of the problem since our algorithm will be then applied to a smaller matrix (i.e., a matrix which contains a subset of columns of the original input matrix). After describing a procedure to maximize such lower bound we then propose a general iterative algorithm that uses the results in Proposition 1 and Corollary 1 to find the right solution.

4.1 A Lower Bound on the Norm

It follows from the analysis presented in Section 3 that only a subset of the columns of the matrix $\boldsymbol{V}$ might be affected by the proximal operator (i.e., those with ℓ_1 norm larger than $t^\star$). This fact can be exploited to reduce the search space of the problem provided that some knowledge about the value of $t^\star$ is available. In particular, having a lower bound on $t^\star$ would allow us to discard columns with smaller ℓ_1 norm. It turns out that a simple lower bound can be derived from the optimality conditions as stated in the following result:

Lemma 2 (Lower-bound on the norm). *Let* $\boldsymbol{X}^\star = \operatorname{prox}_{\lambda\|\cdot\|_{1,\infty}}(\boldsymbol{V})$ *for some* $\boldsymbol{V} \in \mathbb{R}^{n\times m}$. *Let* $t^\star = \|\boldsymbol{X}^\star\|_{1,\infty}$ *be the mixed* $\ell_{1,\infty}$ *norm of the optimal solution. Then, for any subset* $\mathcal{M} \subseteq \{1,\dots,m\}$

$$t_{\mathcal{M}} = \frac{1}{|\mathcal{M}|}\left(\sum_{i\in\mathcal{M}} \|\boldsymbol{v}_i\|_1 - n\lambda\right) \leq t^\star. \tag{39}$$

Proof. From the optimality conditions of Problem (19) we know that $\boldsymbol{x}_i^\star = [\boldsymbol{u}_i - \lambda\mu_i^\star \mathbf{1}]_+$, with $\boldsymbol{u}_i = |\boldsymbol{v}_i|$, and that $\mathbf{1}^\mathsf{T}\boldsymbol{x}_i^\star \leq t^\star$. Then, it follows that

$$\begin{aligned} t^\star &\geq \frac{1}{|\mathcal{M}|}\sum_{i\in\mathcal{M}} \mathbf{1}^\mathsf{T}\boldsymbol{x}_i^\star = \frac{1}{|\mathcal{M}|}\sum_{i\in\mathcal{M}} \mathbf{1}^\mathsf{T}[\boldsymbol{u}_i - \lambda\mu_i^\star\mathbf{1}]_+ \\ &\geq \frac{1}{|\mathcal{M}|}\sum_{i\in\mathcal{M}} \mathbf{1}^\mathsf{T}(\boldsymbol{u}_i - \lambda\mu_i^\star\mathbf{1}) = \frac{1}{|\mathcal{M}|}\sum_{i\in\mathcal{M}} (\mathbf{1}^\mathsf{T}\boldsymbol{u}_i - n\lambda\mu_i^\star) \\ &\geq \frac{1}{|\mathcal{M}|}\left(\sum_{i\in\mathcal{M}} \mathbf{1}^\mathsf{T}\boldsymbol{u}_i - n\lambda\right) = \frac{1}{|\mathcal{M}|}\left(\sum_{i\in\mathcal{M}} \|\boldsymbol{v}_i\|_1 - n\lambda\right), \end{aligned} \tag{40}$$

where the last inequality follows from the fact that $\sum_{i\in\mathcal{M}} \mu_i^\star \leq 1$. □

In order to reduce the search space of the problem, we can maximize the lower bound $t_{\mathcal{M}}$ in (39) with respect to $\mathcal{M}$. Since the sum $\sum_{i\in\mathcal{M}} \|\boldsymbol{v}_i\|_1$ is maximized when we choose the columns of $\boldsymbol{V}$ with the largest ℓ_1 norm, a simple method to compute the set $\mathcal{M}$ that maximizes $t_{\mathcal{M}}$ is to sort the columns of $\boldsymbol{V}$ according to their ℓ_1 norm, evaluate the objective for the top k columns, and choose the value of k that maximizes $t_{\mathcal{M}}$, as described in Algorithm 1. Specifically, we form the vector $\boldsymbol{w}$ that contains the ℓ_1 norms of the columns of $\boldsymbol{V}$ in decreasing order. From $\boldsymbol{w}$ we compute the partial sums $s_k = \sum_{i=1}^k w_i$ for $k = 1,\dots,m$, and evaluate the value of the bound (39) as described in Algorithm 1 to find its maximizer.

Algorithm 1. Maximizing the Lower Bound (39) on $t^\star$

1: Input: $(\boldsymbol{V}, \lambda)$
2: Initialization: $\boldsymbol{U} \leftarrow |\boldsymbol{V}|$ and $\boldsymbol{v} \leftarrow \mathbf{1}^\mathsf{T}\boldsymbol{U}$
3: Sort by ℓ_1 norm such that $w_1 \geq w_2 \geq \cdots \geq w_m$

$$\boldsymbol{w} \leftarrow \operatorname{sort}(\boldsymbol{v})$$

4: Compute the maximizer

$$s_k \leftarrow \sum_{i=1}^k w_i,\ k = 1,\dots,m$$
$$t \leftarrow \max_{1\leq k\leq m} \left((s_k - n\lambda)/k\right)$$

5: Return: t

Note however, that while Algorithm 1 allows us to efficiently find a maximum lower bound $t_{\mathcal{M}}$ for $t^\star$, depending

on the parameter λ, the maximum lower bound might be smaller than zero, in which case it is not useful. In such case, an alternative lower bound for $t^\star$ is given by the following result:

Lemma 3. *Let $\boldsymbol{X}^\star = \text{prox}_{\lambda\|\cdot\|_{1,\infty}}(\boldsymbol{V})$ for some $\boldsymbol{V} \in \mathbb{R}^{n\times m}$. Let $t^\star = \|\boldsymbol{X}^\star\|_{1,\infty}$ be the mixed $\ell_{1,\infty}$ norm of the optimal solution. Then, it holds that*

$$\frac{1}{m}\left(\|\boldsymbol{V}\|_{\infty,1} - \lambda\right) = \frac{1}{m}\left(\sum_{i=1}^{m}\|\boldsymbol{v}_i\|_\infty - \lambda\right) \leq t^\star, \tag{41}$$

Proof. From (25) we know that $\sum_{i=1}^m \mu_i^\star = 1$. It also holds that $t^\star \geq \mathbf{1}^\mathsf{T}\boldsymbol{x}_i^\star$, hence

$$\begin{aligned} m\,t^\star &\geq \sum_{i=1}^{m}\mathbf{1}^\mathsf{T}\boldsymbol{x}_i^\star = \sum_{i=1}^{m}\mathbf{1}^\mathsf{T}[\boldsymbol{u}_i - \lambda\mu_i^\star\mathbf{1}]_+ \\ &\geq \sum_{i=1}^{m}\left(\max_{1\leq j\leq n} u_{ij} - \lambda\mu_i^\star\right) = \sum_{i=1}^{m}\|\boldsymbol{u}_i\|_\infty - \lambda\sum_{i=1}^{m}\mu_i^\star \\ &= \|\boldsymbol{U}\|_{\infty,1} - \lambda = \|\boldsymbol{V}\|_{\infty,1} - \lambda. \end{aligned} \tag{42}$$

□

Note that the bound in (41) will be negative only if the optimal solution is the zero matrix since:

$$\begin{aligned} \|\boldsymbol{V}\|_{\infty,1} < \lambda &\implies \text{P}_{\|\cdot\|_{\infty,1}\leq\lambda}(\boldsymbol{V}) = \boldsymbol{V} \\ &\implies \text{prox}_{\lambda\|\cdot\|_{1,\infty}}(\boldsymbol{V}) = \boldsymbol{0}. \end{aligned} \tag{43}$$

4.2 A General Algorithm

A general procedure for computing the proximal operator of the mixed $\ell_{1,\infty}$ norm can be devised based on the optimality conditions of Proposition 1 and the observation that, for a fixed t, the problem in (19) boilds down to projecting the columns of $\boldsymbol{U}$ onto the ℓ_1 ball of radius t. A possible strategy for finding $t^\star$ is to start with a lower bound t for $t^\star$, project each column of U whose ℓ_1 norm is above the current lower bound onto the ℓ_1 ball of radius t, update the value of the lower bound using (21), and keep iterating until there are no further changes in t (see Algorithm 2). This algorithm is guaranteed to converge to the optimal solution, as stated next.

Algorithm 2. Proximal Operator of Mixed $\ell_{1,\infty}$ Norm: $\text{prox}_{\lambda\|\cdot\|_1}(\boldsymbol{V})$

1: Initialization: $\boldsymbol{U} \leftarrow |\boldsymbol{V}|$
2: Compute lower bound on t
3: **do**
4: $\quad\mathcal{M}$-update: $\mathcal{M} \leftarrow \{i \,|\, t < \|\boldsymbol{u}_i\|_1\}$
5: $\quad$**for** $i \in \mathcal{M}$ **do**
6: $\quad\quad$Projection onto the simplex: $\boldsymbol{x}_i \leftarrow \text{P}_{\|\cdot\|_1\leq t}(\boldsymbol{u}_i)$
7: $\quad\quad\mathcal{J}_i$-update: $\mathcal{J}_i \leftarrow \{j \,|\, x_{ij} > 0\}$
8: $\quad$**end for**
9: $\quad t$-update: $t = \frac{\sum_{i\in\mathcal{M}}\frac{1}{|\mathcal{J}_i|}\sum_{j\in\mathcal{J}_i} u_{ij} - \lambda}{\sum_{i\in\mathcal{M}}\frac{1}{|\mathcal{J}_i|}}$
10: **while** $\mathcal{M}$ or $\{\mathcal{J}_i\}_{i=1}^m$ change
11: Compute proximal operator using Corollary 1.

Proposition 2 (Correctness). *Algorithm 2 converges to the proximal operator of the mixed $\ell_{1,\infty}$ norm of matrix $\boldsymbol{V} \in \mathbb{R}^{n\times m}$ in at most nm iterations.*

Proof. We prove that the algorithm reaches the optimal solution by showing that starting from a lower bound, the method produces a monotonic sequence of values for t that eventually converges to the optimal value $t^\star$. To see this, note that for a given t, the projection onto the ℓ_1 ball has the form of (30) that is, $\boldsymbol{x}_i = \left[\boldsymbol{u}_i - \lambda\mu_i\mathbf{1}\right]_+$ for some value μ_i. Let $\mathcal{M}$ be the set of columns with ℓ_1 norm larger than t (i.e., $\mathcal{M} = \{i \,|\, \|\boldsymbol{u}_i\|_1 > t\}$), and let $\mathcal{J}_i$, $i \in \mathcal{M}$ denote the sets of non-zero entries for the ith column after projecting onto the ℓ_1 ball of radius t. Let also $\lambda\mu_i^\star$ denote the (per column) thresholding values at the optimal solution. Now since t is a lower bound on $t^\star$ then it is necessary the case that $\mathcal{M}^\star \subseteq \mathcal{M}$ and also that $\mu_i \geq \mu_i^\star$ (hence $\mathcal{J}_i \subseteq \mathcal{J}_i^\star$), where $\mathcal{M}^\star$ is the subset of columns that are being thresholded at the optimal solution. With these considerations in mind, we can now compute the new value t^+ using the t-update step 14 in Algorithm 2 to get

$$\begin{aligned} t^+ &= \frac{\sum_{i\in\mathcal{M}}\frac{1}{|\mathcal{J}_i|}\sum_{j\in\mathcal{J}_i} u_{ij} - \lambda}{\sum_{i\in\mathcal{M}}\frac{1}{|\mathcal{J}_i|}} \\ &= \frac{\sum_{i\in\mathcal{M}}\frac{1}{|\mathcal{J}_i|}\sum_{j\in\mathcal{J}_i}\left(u_{ij} - \mu_i\lambda + \mu_i\lambda\right) - \lambda}{\sum_{i\in\mathcal{M}}\frac{1}{|\mathcal{J}_i|}} \\ &= \frac{\sum_{i\in\mathcal{M}}\frac{1}{|\mathcal{J}_i|}t + \mu_i\lambda - \lambda}{\sum_{i\in\mathcal{M}}\frac{1}{|\mathcal{J}_i|}} = t + \lambda\frac{\sum_{i\in\mathcal{M}}\mu_i - 1}{\sum_{i\in\mathcal{M}}\frac{1}{|\mathcal{J}_i|}} \geq t, \end{aligned} \tag{44}$$

where the last inequality follows from the fact that $\mu_i \geq \mu_i^\star$, $\sum_{i\in\mathcal{M}^\star}\mu_i^\star = 1$ (see (25)) and $\mathcal{M}^\star \subseteq \mathcal{M}$. This proves that starting from a lower bound of t, an iteration of Algorithm 2 produces a new value that is larger or equal than the previous one. If the new value is equal to the previous one (i.e., $t^+ = t$), it means we have found the optimal solution since the optimality conditions of Proposition 1 are satisfied (note that the sets $\mathcal{M}$ and $\{\mathcal{J}_i\}_{i=1}^m$ are determined by t and no changes in the value of t would imply no changes in those sets either). For the case where $t^+ > t$, we need to show that the new value t^+ remains a lower bound on the optimal value. This is the case since the t-update corresponds to the minimizer of the cost function (19) given that we fix the support of $\boldsymbol{X}$ (i.e., specify the sets $\mathcal{M}$ and $\{\mathcal{J}_i\}_{i=1}^m$). We can easily see this by plugging the solution $\boldsymbol{X}^\star$ as a function of t into the cost function to get the function:

$$f(t) = t + \frac{1}{2\lambda}\sum_{i=1}^{m}\|\boldsymbol{x}_i^* - \boldsymbol{u}_i\|_2^2 = t + \frac{1}{2\lambda}\sum_{i\in\mathcal{M}}(\lambda\mu_i)^2. \tag{45}$$

Substituting the value of μ_i in (22) into (45) we get

$$f(t) = t + \frac{1}{2\lambda}\sum_{i\in\mathcal{M}}\left(\frac{\sum_{j\in\mathcal{J}_i} - t}{|\mathcal{J}_i|}\right)^2. \tag{46}$$

Differentiating (46) with respect to t and equating to zero yields the update equation for t. Since the sets $\mathcal{M}$ and $\mathcal{J}_i$ are determined based on a lower bound of $t^\star$ then the quadratic term in the optimization problem (19) is larger than its value at the optimal threshold $t^\star$. Therefore, the updated t value must be smaller that $t^\star$. As a consequence, starting from a lower bound $t^{(0)}$ the algorithm

TABLE 1
Average Execution Time and Average Number of Iterations (in Brackets) of the Different Methods in Computing the Projection Onto the $\ell_{\infty,1}$ Ball

Size	α	ST [13]	NT [19]	QT [8]	Sort	Active Set	Bisection
100×100	10^{-4}	1.48E-03 [2.0]	7.54E-04 [2.4]	2.12E-03 [-]	1.51E-04 [1.9]	**1.26E-04** [2.8]	1.72E-04 [49.1]
	10^{-3}	3.66E-03 [6.5]	2.01E-03 [6.5]	2.19E-03 [-]	1.80E-04 [2.1]	**1.09E-04** [4.2]	1.55E-04 [67.1]
	10^{-2}	6.83E-03 [8.3]	4.37E-03 [8.3]	1.98E-03 [-]	4.34E-04 [2.8]	**1.36E-04** [6.8]	4.07E-04 [85.5]
	10^{-1}	6.62E-03 [7.2]	4.40E-03 [7.2]	1.64E-03 [-]	6.27E-04 [3.1]	**2.43E-04** [9.8]	1.73E-03 [103.2]
1000×100	10^{-4}	1.38E-02 [8.1]	6.08E-03 [8.1]	1.72E-02 [-]	7.90E-04 [2.1]	**6.31E-04** [4.1]	7.18E-04 [68.6]
	10^{-3}	3.42E-02 [10.7]	1.96E-02 [10.7]	1.61E-02 [-]	1.05E-03 [2.7]	**6.36E-04** [5.8]	9.99E-04 [81.3]
	10^{-2}	4.32E-02 [9.0]	2.89E-02 [9.0]	2.02E-02 [-]	3.43E-03 [3.1]	**1.08E-03** [9.1]	1.83E-03 [107.3]
	10^{-1}	9.26E-02 [7.9]	6.31E-02 [7.9]	3.87E-02 [-]	1.22E-02 [3.8]	**5.08E-03** [14.2]	5.58E-03 [117.7]
100×1000	10^{-4}	3.99E-03 [4.0]	2.59E-03 [4.1]	1.81E-02 [-]	9.95E-04 [2.0]	**6.70E-04** [4.0]	1.19E-03 [68.8]
	10^{-3}	1.07E-02 [6.7]	6.75E-03 [6.7]	1.72E-02 [-]	2.04E-03 [2.3]	**6.88E-04** [5.1]	2.32E-03 [79.2]
	10^{-2}	2.45E-02 [8.6]	1.46E-02 [8.6]	1.65E-02 [-]	6.25E-03 [3.0]	**1.06E-03** [9.0]	9.41E-03 [107.0]
	10^{-1}	2.33E-02 [8.0]	1.47E-02 [8.0]	1.67E-02 [-]	7.25E-03 [3.8]	**2.46E-03** [14.2]	2.19E-02 [140.5]
1000×1000	10^{-4}	9.97E-02 [8.8]	6.57E-02 [8.8]	2.01E-01 [-]	1.05E-02 [2.3]	**7.91E-03** [5.0]	1.01E-02 [76.5]
	10^{-3}	2.66E-01 [11.0]	1.71E-01 [11.0]	2.00E-01 [-]	2.20E-02 [2.9]	**8.57E-03** [8.6]	1.83E-02 [99.8]
	10^{-2}	3.14E-01 [9.0]	2.16E-01 [9.0]	2.00E-01 [-]	6.53E-02 [3.1]	**1.38E-02** [9.0]	7.35E-02 [109.6]
	10^{-1}	3.05E-01 [8.2]	2.15E-01 [8.2]	1.99E-01 [-]	7.52E-02 [4.0]	**3.16E-02** [16.0]	2.30E-01 [151.2]
10000×1000	10^{-4}	3.55E+00 [13.0]	2.23E+00 [13.0]	2.62E+00 [-]	1.16E-01 [3.0]	**9.60E-02** [8.3]	1.14E-01 [100.7]
	10^{-3}	3.62E+00 [11.0]	2.38E+00 [11.0]	2.61E+00 [-]	2.23E-01 [3.0]	**1.10E-01** [9.0]	1.86E-01 [112.1]
	10^{-2}	4.41E+00 [9.0]	2.99E+00 [9.0]	2.64E+00 [-]	6.57E-01 [3.7]	**1.71E-01** [9.5]	2.21E-01 [115.2]
	10^{-1}	4.54E+00 [8.8]	3.09E+00 [8.8]	2.65E+00 [-]	7.59E-01 [4.0]	**3.51E-01** [16.0]	3.98E-01 [159.8]

The results correspond to an average over 100 realizations.

produces a sequence of the form:

$$t^{(0)} < t^{(1)} < \ldots < t^{(k)} = t^{\star}, \tag{47}$$

with corresponding sets $\mathcal{M}^{(0)} \supseteq \mathcal{M}^{(1)} \supseteq \ldots \supseteq \mathcal{M}^{\star}$ and $\mathcal{J}_i^{(0)} \subseteq \mathcal{J}_i^{(1)} \subseteq \ldots \subseteq \mathcal{J}_i^{\star}$. Now since a change on t necessarily implies a change in the support sets $\mathcal{M}$ and/or $\{\mathcal{J}_i\}_{i=1}^m$ and due to the inclusion relationships over iterations at least one new element of the non-zero support of the solution is added at every iteration. Since the non-zero support of the solution has at most mn elements, the algorithm terminates in at most mn iterations. □

4.3 Particular Implementations and Complexity

The complexity of Algorithm 2 depends on the method used for computing the projection step onto the simplex and there exist different alternatives in the literature [16]. A naive implementation of the proposed algorithm can lead to a computationally inefficient method if at every iteration the projection step is computed from scratch. Alternatively, one could exploit previous estimates from one iteration to the next in order to improve the computational efficiency. In this paper we explore two different approaches for computing the projection step onto the simplex. The first one is based on sorting the columns of the matrix of absolute values that are affected by thresholding as described in Algorithm 3. In such case, the expected complexity of the method is dominated by the sorting operation and it is $O(mn\log n)$ operations. Note that the projection onto the simplex step only needs to be updated from one iteration to the next due to the monotonicity of the t sequence. We can think of the method as a nested version of [17] where the ball radius t is updated in the outer loop allowing for a warm start of the projection step. The second approach is based on an active set method based on Proposition 1 as described in Algorithm 4. In this case, the nested projection onto the simplex is equivalent to Michelot's method [18]. While in the latter case the worst case complexity of the projection step for each column is $O(n^2)$ [16] we have empirically observed that the method is more efficient than the sorting-based one. Since both Algorithms 3 and 4 are particular implementations of Algorithm 2 they are guaranteed to terminate in at most mn (outer) iterations. In practice, however we have observed that the number of iterations to reach convergence is much smaller (see Table 1).

Algorithm 3. Sorting-Based $\operatorname{prox}_{\lambda\|\cdot\|_1}(V)$

Initialization: $U \leftarrow |V|$
`#==== Pre-processing ====#`
Compute lower bound on t
$\mathcal{M}$ initialization: $\mathcal{M} \leftarrow \{i \,|\, t < \|\boldsymbol{u}_i\|_1\}$
for $i \in \mathcal{M}$ **do**
  Sort $\boldsymbol{u}_i$ into $\boldsymbol{s}_i$: $s_{i1} \geq s_{i2} \geq \ldots \geq s_{in}$
  $\mathcal{J}_i = \{s_{i1}\}$ `#Initialize` $\mathcal{J}_i$
end for
`#==== Iterations ========#`
do
  for $i \in \mathcal{M}$ **then**
    while $|\mathcal{J}_i| < n$
      if $\big(\sum_{j=1}^{|\mathcal{J}_i|} s_{ij} - t\big)/|\mathcal{J}_i| < s_{i|\mathcal{J}_i|}$ **then**
        $\mathcal{J}_i = \mathcal{J}_i \cup \{j \,|\, s_{i|\mathcal{J}_i|} = x_{ij}\}$ `#Lookup table`
      else
        break
  end for
  t-update: $t = \frac{\sum_{i\in\mathcal{M}} \frac{1}{|\mathcal{J}_i|} \sum_{j\in\mathcal{J}_i} u_{ij} - \lambda}{\sum_{i\in\mathcal{M}} \frac{1}{|\mathcal{J}_i|}}$
  $\mathcal{M}$-update: $\mathcal{M} \leftarrow \{i \,|\, t < \|\boldsymbol{u}_i\|_1\}$
while $\mathcal{M}$ or $\{\mathcal{J}_i\}_{i=1}^m$ change
Compute proximal operator using Corollary 1.

Algorithm 4. Active Set $\text{prox}_{\lambda\|\cdot\|_1}(\boldsymbol{V})$

Initialization: $\boldsymbol{U} \leftarrow |\boldsymbol{V}|$
#==== Pre-processing ====#
Compute lower bound on t
$\mathcal{M}$ initialization: $\mathcal{M} \leftarrow \{i \,|\, t < \|\boldsymbol{u}_i\|_1\}$
Partial sums: $S_i^{(0)} = \|\boldsymbol{u}_i\|_1,\ i = 1, \ldots, m.$
#==== Iterations ========#
do
  for $i \in \mathcal{M}$ **do**
    $S_i \leftarrow S_i^{(0)},\ \mathcal{J}_i \leftarrow \{1, \ldots, n\}$ #Initialize
  do
    $\mu_i = (S_i - t)/(\lambda|\mathcal{J}_i|)$
    for $j = 1, \ldots, n$ **do**
      if $u_{ij} \leq \mu_i$ **then**
        $\mathcal{J}_i = \mathcal{J}_i - \{j\}$
        $S_i = S_i - u_{ij}$
      end for
    while $\mathcal{J}_i$ changes
  end for
  t-update: $t = \frac{\sum_{i\in\mathcal{M}} \frac{1}{|\mathcal{J}_i|} \sum_{j\in\mathcal{J}_i} u_{ij} - \lambda}{\sum_{i\in\mathcal{M}} \frac{1}{|\mathcal{J}_i|}}$
  $\mathcal{M}$-update: $\mathcal{M} \leftarrow \{i \,|\, t < \|\boldsymbol{u}_i\|_1\}$
while $\mathcal{M}$ or $\{\mathcal{J}_i\}_{i=1}^m$ change
Compute proximal operator using Corollary 1.

4.3.1 Bisection Search

The algorithms presented so far approach the solution from below, starting with a lower bound on the optimal value t. Given an initial search interval that contains the optimal solution $t^\star$ and, using the update equation for the value of t it is possible to devise a bisection procedure that would iteratively shrink the search interval. A lower bound is given by Lemmas 2 and 3. A trivial upper bound is given by the $\ell_{1,\infty}$ norm of $\boldsymbol{V}$. From (44) we know that starting from a lower bound, the t-update will necessarily be larger or equal than the previous value. Furthermore, it will only be equal provided we are at the optimal solution. Likewise, by starting with an upper bound of t the update equation will lead to a less than or equal value for t. In order to see this, recall from (44) that:

$$t^+ = t + \lambda \frac{\sum_{i\in\mathcal{M}} \mu_i - 1}{\sum_{i\in\mathcal{M}} \frac{1}{|\mathcal{J}_i|}}.$$

Now, we can have two different situations:

- $t < t^\star$ is a *lower bound* which implies that $\mathcal{M} \supseteq \mathcal{M}^\star$ and $\mu_i > \mu_i^\star$. Therefore, $\sum_{i\in\mathcal{M}} \mu_i > \sum_{i\in\mathcal{M}} \mu_i^\star = 1$ and hence $t^+ > t$ (see (44)).
- $t > t^\star$ is an *upper bound* which implies that $\mathcal{M} \subseteq \mathcal{M}^\star$ and $\mu_i < \mu_i^\star$. Therefore, $\sum_{i\in\mathcal{M}} \mu_i < \sum_{i\in\mathcal{M}} \mu_i^\star \leq \sum_{i\in\mathcal{M}} \mu_i^\star = 1$ and hence $t^+ < t$.

As a consequence of the above relations a simple bisection search can be devised for finding an ϵ-optimal value for t as summarized in Algorithm 5. Note that the algorithm uses the fact that starting from a lower bound the t-update step results in yet another lower bound, so the updated lower bound is set to the newly computed value t^+ as opposed to t. Starting from an upper bound it is not necessarily the case that the new updated value remains an upper bound, hence the update for the upper bound uses the tested t value instead.

Algorithm 5. Bisection $\text{prox}_{\lambda\|\cdot\|_1}(\boldsymbol{V})$

Set tolerance: $\epsilon \geq 0$
Initialization: $\boldsymbol{U} \leftarrow |\boldsymbol{V}|$
#==== Pre-processing ====#
Compute lower bound t_L
Compute upper bound $t_U = \max_i \mathbf{1}^T \boldsymbol{u}_i$
Inital test point: $t = t_L$
#==== Iterations ========#
while $(t_U - t_L > \epsilon)$
  $\mathcal{M}$-update: $\mathcal{M} \leftarrow \{i \,|\, t < \|\boldsymbol{u}_i\|_1\}$
  for $i \in \mathcal{M}$ **do**
    Projection onto the simplex: $\boldsymbol{x}_i \leftarrow \mathrm{P}_{\|\cdot\|_1 \leq t}(\boldsymbol{u}_i)$
    $\mathcal{J}_i$-update: $\mathcal{J}_i \leftarrow \{j \,|\, x_{ij} > 0\}$
  end for
  t-update: $t^+ = \frac{\sum_{i\in\mathcal{M}} \frac{1}{|\mathcal{J}_i|} \sum_{j\in\mathcal{J}_i} u_{ij} - \lambda}{\sum_{i\in\mathcal{M}} \frac{1}{|\mathcal{J}_i|}}$
  if $t^+ > t$ **then**
    $t_L = t^+$ #New lower bound
  else
    $t_U = t$ #New upper bound
  end if
  New test point: $t = (t_U + t_L)/2$
end while
Compute $\text{prox}_{\lambda\|\cdot\|_1}(\boldsymbol{V})$ using Corollary 1 based on t_L.

5 Numerical Experiments

5.1 Numerical Validation

In order to evaluate the computational complexity of the proposed algorithms we randomly generate matrices in $\mathbb{R}^{n\times m}$ with independent and identically distributed random entries drawn from a uniform distribution $\mathcal{U}([-0.5, 0.5])$. We then apply the proposed implementations of Algorithm 2 to compute projections onto the mixed $\ell_{\infty,1}$ ball for different values of the ball radius. We label our implementations as "Sort" for the sorting-based implementation, "Active Set" for the one based on active sets, and "Bisection" for the one based on a bisection search. For the bisection approach, we use the same projection method as in the Active Set method of Algorithm 4. We also compute the projections using the state of the art algorithms. In particular we compare to the method proposed in [8] which we denote as "QT" and with the recently proposed root-search based methods of [13], [19] which we denote as "ST" (Steffensen) and "NT" (Newton), respectively. We record the execution time for different configurations (sizes) of the data matrix and for different values of the $\ell_{\infty,1}$ ball radius. In our experiments, we choose the radius of the ball to be a fraction $\alpha \in [0, 1]$ of the true mixed norm of the matrix and compute the average computation time over 100 realizations. For the methods in [8], [13], [19] we use the implementations provided by the authors. The tolerance for root-search based methods is set to its default value of $\epsilon = 1E-10$. For the bisection search method we set the tolerance to that same value. The numerical experiments have been conducted on a 2.8 GHz machine with Intel I7 processor, and using Matlab R2019b. The results for different matrix sizes are displayed in Table 1 where we report the averages for the

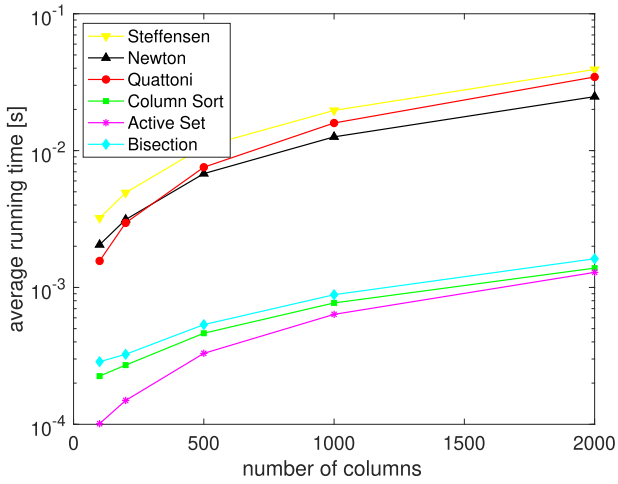

Fig. 2. Average execution time for the projection onto the $\ell_{\infty,1}$ ball for randomly generated matrices of $n = 100$ rows and for different number of columns. The data is projected onto a ball whose radius is a fraction $\alpha = 0.1$ of the original matrix. The results correspond to an average over 100 realizations.

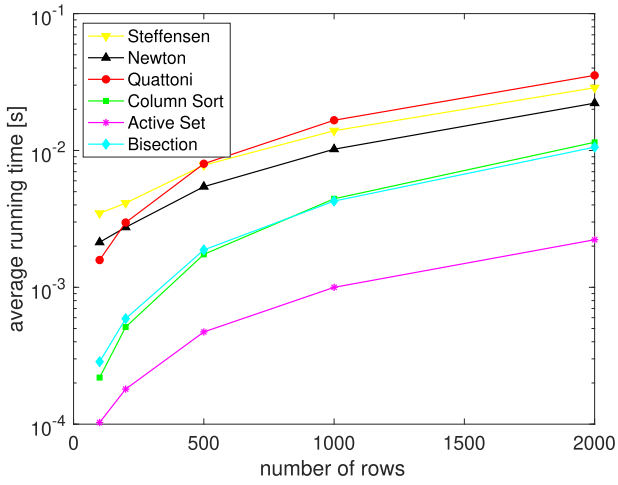

Fig. 3. Average execution time for the projection onto the $\ell_{\infty,1}$ ball for randomly generated matrices with $m = 100$ columns and for different number of rows. The data is projected onto a ball whose radius is a fraction $\alpha = 0.1$ of the original matrix. The results correspond to an average over 100 realizations.

execution time and the number of iterations[3] (in brackets). As it can be observed from the table, our implementations achieve the best performance offering an improvement over the state of the art that ranges between one and two orders of magnitude with the Active Set method being the most efficient in this setting. We also observed that both the Sorting and Active Set methods converge to the true solution in a very small number of iterations, far below to the worst-case case of mn.

Figs. 2 and 3 display the average execution time for the case of 100×100 randomly generated matrices and for a target fraction of the ball radius of $\alpha = 0.1$ as we vary the number of columns (resp. rows). Also, Fig. 4 displays the execution time for the different methods as a function of the fraction of the norm α. As we can see, our proposed approaches significantly outperform all previous methods.

3. For QT the implementation provided by the authors does not return the number of iterations and hence it is not reported in Table 1.

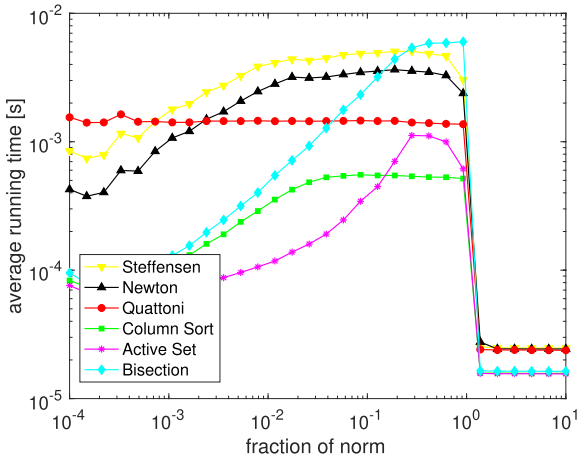

Fig. 4. Average execution time for the projection onto the $\ell_{\infty,1}$ ball for randomly generated matrices of size data 100×100. The data is projected onto a ball whose radius is a fraction α of the original matrix. The results correspond to an average over 100 realizations.

TABLE 2
Characterization Summary of the Used Datasets, see [20]

Dataset	Classes n	Samples p	Dimension m
Carcinom [21], [22]	11	174	9182
GLIOMA [23]	4	50	4434
LUNG [24]	5	203	3312
ALLAML [25]	2	72	7129
Prostate-GE [26]	2	102	5966

In general, we also observe that all methods (except Quattoni's) perform better when projecting onto a ball of small radius and that in such regime the root-search-based methods can outperform Quattonni's. The drop in execution time for $\alpha \geq 1$ corresponds to the case where there is no thresholding (e.g., projection is the identity).

5.2 Application to Cancer Classification From Gene Expression Data

In this section we test our algorithms in the context of multi-task learning for the problem of cancer classification from gene expression data where the dimensionality of the feature vectors m is typically much larger than the number of samples p. We use the datasets provided in [20] which consist of five curated datasets of different types of cancers as described in [20]. The datasets are briefly summarized in Table 2. We pose the classification problem as a multi-task learning problem. In particular, given a dataset of points with associated labels $\mathcal{D} = \{(\boldsymbol{x}_i, c_i)\}_{i=1}^{p}$, with $\boldsymbol{x}_i \in \mathbb{R}^m$ and $c_i \in \{0, \ldots, n\}$, where n is the number of classes, we build a data matrix $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_p]^T$ and target label matrix $\boldsymbol{Y} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_p]^T$ with

$$\boldsymbol{y}_i = [y_{i1}, \ldots, y_{in}]^{\mathsf{T}}, \quad y_{ij} = \begin{cases} 1 & j = c_i \\ 0 & \text{else} \end{cases}. \tag{48}$$

The problem is to predict the correct label for each class while enforcing feature sharing among them:

$$\begin{aligned} \underset{\boldsymbol{W}}{\text{minimize}} \quad & \|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{W}^T\|_F^2 \\ \text{subject to} \quad & \|\boldsymbol{W}\|_{\infty,1} \leq \tau \end{aligned}. \tag{49}$$

TABLE 3
Average Classification Accuracy Using Criterion (50)

Dataset	Carcinom [11 classes]	GLIOMA [4 classes]	LUNG [5 classes]	ALLAML [2 classes]	Prostate-GE [2 classes]
$\ell_{2,1}$ (Nie *et al.*)	95.50	68.90	76.95	92.36	93.25
$\ell_{\infty,1}$ (Proposed)	**97.74**	**78.50**	**83.28**	**95.07**	**93.65**

Note that problem (49) falls within the family of problems in (2) which can be solved using a projected gradient descent strategy. For the projection step onto the $\ell_{\infty,1}$ ball we use the sorting-based implementation of Algorithm 2.

We conducted an experiment using the datasets of Table 2 where we center the data points (mean subtraction) and normalize them by dividing each coordinate by its standard deviation. For each dataset we split the data into 80 percent training and 20 percent testing and computed the average classification performance over 100 random data splits. Once we solve (49) we use the following simple classification rule:

$$\hat{c}_i = \underset{1\leq j\leq n}{\operatorname{argmax}}\ \hat{y}_{ij}, \quad \hat{\boldsymbol{Y}} = \boldsymbol{X}\boldsymbol{W}^T = [\hat{\boldsymbol{y}}_1, \ldots, \hat{\boldsymbol{y}}_p]^T. \tag{50}$$

In addition, we use the learned weights to identify relevant features and train a (kernel) support vector machine (SVM) classifier on the identified features. Features are sorted according to the euclidean norm of the columns of $\boldsymbol{W}$ being the most relevant index the one with larger norm. For the multi-class problem we use a *one-versus-one* strategy with majority voting. We also provide a comparison with the $\ell_{2,1}$ norm based feature selection method of [20] for which we used the implementation provided by the authors. The $\ell_{\infty,1}$ ball radius τ in (49) as well as the regularization parameter for the method in [20] were chosen using a grid search. The average classification accuracy of both methods the classification rule (50) are summarized in Table 3. As it can be appreciated we observe that the proposed method using the $\ell_{\infty,1}$ norm achieves better classification accuracy than the method based on the $\ell_{2,1}$ proposed in [20]. It is important to note that the differences are more pronounced in multi-class problems than in binary ones indicating as expected, that the $\ell_{\infty,1}$ norm encourages the discovery of variables that are most correlated.

We also report the classification results using an SVM classifier and for different number of features used. The results are displayed in Fig. 6 for all datasets. We can observe the superior performance of the proposed scheme in selecting relevant features for the discrimination task. Again the performance gap is generally more pronounced on those datasets with more than two classes.

Finally, we also provide a comparison between the execution time required for the computation of the two steps in the projected gradient descent. In particular, we record the execution time for the computation of the gradient and compare it with the time it takes to project onto the $\ell_{\infty,1}$ ball. We do this for the largest (Carcinom) dataset. We display such projection time for all the considered projection methods in this paper. The results are illustrated in Fig. 5. Again, we appreciate the improved efficiency of our proposed algorithms as compared to the state of the art. We can also see that the computation of the projection operator using our proposed methods can be cheaper than the computation of the gradient step. Surprisingly the bisection search appears to be the most efficient in this real data scenario. A possible reason for such behavior might be associated to the existing correlation among the data variables. In any case, the bisection search can be a suitable option when there is a need to trade-off between accuracy and computational complexity.

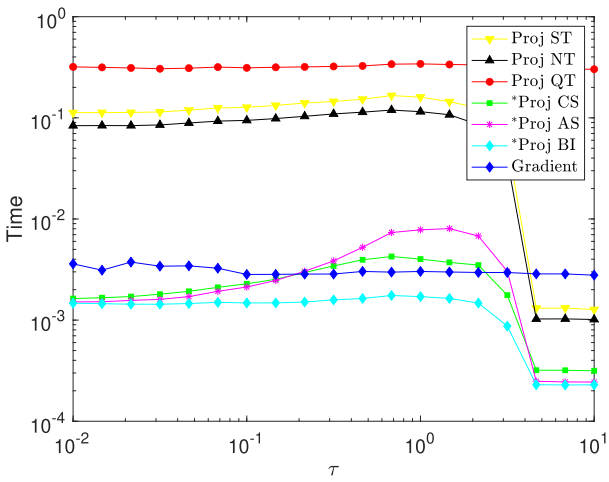

Fig. 5. Average execution time for the projection onto the $\ell_{\infty,1}$ ball for the multi-task learning problem using the Carcinom dataset as compared to the computation time for the gradient step. We report the results for different values of the $\ell_{\infty,1}$ ball radius τ. The results correspond to an average over 5 random data splits where 80 percent of the data is used for training. Our methods are indicated by an asterisk * where CS, AS, and BI denote "Column Sort", "Active Set" and "Bisection," respectively.

6 Conclusion

In this paper we have analyzed in detail the proximity operator of the mixed $\ell_{1,\infty}$ matrix norm. We have provided simple expressions for its computation that generalize the well-known soft-thresholding algorithm. By exploiting the duality relationship to the $\ell_{\infty,1}$ norm we also derive the projection operator onto the mixed $\ell_{\infty,1}$ norm. In addition, we have proposed a general algorithm for the computation of the proximal operator and two particular implementations that can be orders of magnitude faster than the state of the art making them particularly suitable for large-scale problems. We have also illustrated the application of the $\ell_{\infty,1}$ norm for biomarker discovery (feature selection) for the problem of cancer classification from gene expression data.

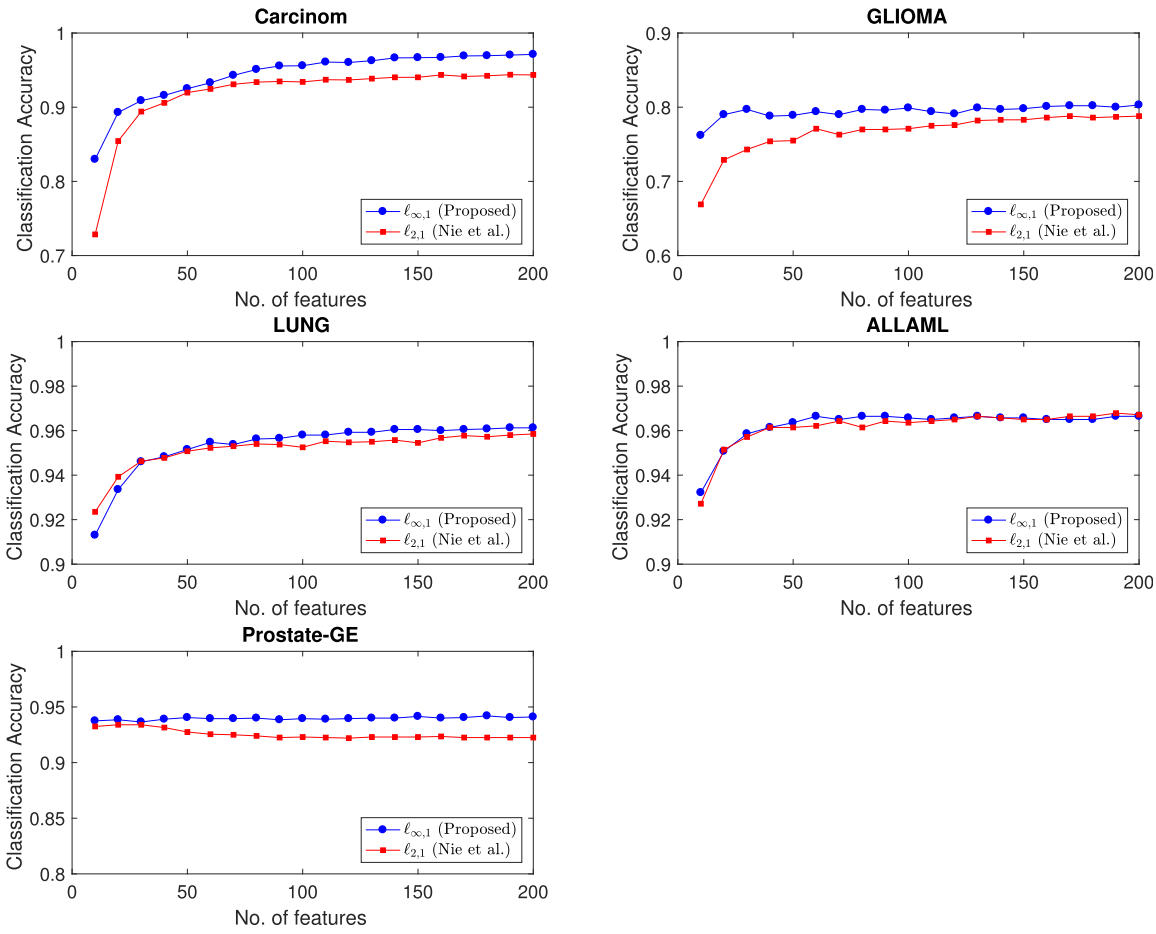

Fig. 6. Average classification results using an SVM classifier as a function of the number of features selected.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their suggestions and constructive comments that have certainly contributed to improve the quality of this paper. This work acknowledges financial support under grant NSF 1704458.

REFERENCES

[1] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. New York, NY, USA: Springer, 2006.

[2] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY, USA: Cambridge Univ. Press, 2004.

[3] P. L. Combettes and J.-C. Pesquet, "Proximal splitting methods in signal processing," in *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, H. Bauschke, R. Burachik, P. Combettes, V. Elser, D. Luke, and H. Wolkowicz, Eds. Berlin, Germany: Springer, 2011, pp. 185–212. [Online]. Available: https://hal.inria.fr/hal-00643807

[4] F. Bach, R. Jenatton, J. Mairal, and G. Obozinski, "Optimization with sparsity-inducing penalties," *Foundations Trends Mach. Learn.*, vol. 4, no. 1, pp. 1–106, 2012. [Online]. Available: http://dx.doi.org/10.1561/2200000015

[5] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations Trends® Optim.*, vol. 1, no. 3, pp. 127–239, 2014. [Online]. Available: http://dx.doi.org/10.1561/2400000003

[6] G. Obozinski, B. Taskar, and M. I. Jordan, "Joint covariate selection and joint subspace selection for multiple classification problems," *Statist. Comput.*, vol. 20, no. 2, pp. 231–252, 2010. [Online]. Available: http://dx.doi.org/10.1007/s11222-008-9111-x

[7] B. A. Turlach, W. N. Venables, and S. J. Wright, "Simultaneous variable selection," *Technometrics*, vol. 47, no. 3, pp. 349–363, Aug. 2005.

[8] A. Quattoni, X. Carreras, M. Collins, and T. Darrell, "An efficient projection for $l_{1,\infty}$ regularization," in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, 2009, pp. 857–864. [Online]. Available: http://doi.acm.org/10.1145/1553374.1553484

[9] E. Elhamifar, G. Sapiro, and R. Vidal, "See all by looking at a few: Sparse modeling for finding representative objects," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 1600–1607.

[10] I. Dokmanić and R. Gribonval, "Beyond moore-penrose part I: generalized inverses that minimize matrix norms," *CoRR*, vol. abs/1706.08349, 2017. [Online]. Available: http://arxiv.org/abs/1706.08349

[11] I. Dokmanić and R. Gribonval, "Concentration of the frobenius norm of generalized matrix inverses," *SIAM J. Matrix Anal. Appl.*, vol. 40, no. 1, pp. 92–121, 2019.

[12] S. Sra,, "Fast projections onto $\ell_{1,q}$-norm balls for grouped feature selection," in *Machine Learning and Knowledge Discovery in Databases*. Berlin, Germany: Springer, 2011, pp. 305–317.

[13] G. Chau, B. Wohlberg, and P. Rodríguez, "Fast projection onto the $\ell_{\infty,1}$ mixed norm ball using steffensen root search," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2018, pp. 4694–4698.

[14] J. J. Moreau, "Proximité et dualtité dans un espace Hilbertien," *Bulletin de la Societé Mathématique de France*, vol. 93, pp. 273–299, 1965.

[15] D. P. Palomar and J. R. Fonollosa, "Practical algorithms for a family of waterfilling solutions," *IEEE Trans. Signal Process.*, vol. 53, no. 2, pp. 686–695, Feb. 2005.

[16] L. Condat, "Fast projection onto the simplex and the ℓ_1 ball," *Math. Program.*, vol. 158, no. 1, pp. 575–585, 2016. [Online]. Available: http://dx.doi.org/10.1007/s10107-015-0946-6

[17] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra, "Efficient projections onto the ℓ_1-ball for learning in high dimensions," in *Proc. 25th Int. Conf. Mach. Learn.*, 2008, pp. 272–279.

[18] C. Michelot, "A finite algorithm for finding the projection of a point onto the canonical simplex of $\mathbb{R}^n$," *J. Optim. Theory Appl.*, vol. 50, no. 1, pp. 195–200, 1986.
[19] G. Chau, B. Wohlberg, and P. Rodríguez, "Efficient projection onto the $\ell_{\infty,1}$ mixed-norm ball using a newton root search method," *SIAM J. Imaging Sci.*, vol. 12, pp. 604–623, 2019.
[20] F. Nie, H. Huang, X. Cai, and C. H. Ding, "Efficient and robust feature selection via joint $\ell_{2,1}$-norms minimization," in *Proc. 23rd Int. Conf. Neural Inf. Process. Syst.*, 2010, pp. 1813–1821.
[21] A. I. Su *et al.*, "Molecular classification of human carcinomas by use of gene expression signatures," *Cancer Res.*, vol. 61, no. 20, pp. 7388–7393, 2001. [Online]. Available: http://cancerres.aacrjournals.org/content/61/20/7388
[22] K. Yang, Z. Cai, J. Li, and G. Lin, "A stable gene selection in microarray data analysis," *BMC Bioinf.*, vol. 7, pp. 228–243, 2006. [Online]. Available: http://search.ebscohost.com.proxy1.library.jhu.edu/login.aspx?direct=true&db=asn&AN=43698712&site=ehost-live&scope=site
[23] C. L. Nutt *et al.*, "Gene expression-based classification of malignant gliomas correlates better with survival than histological classification," *Cancer Res.*, vol. 63, no. 7, pp. 1602–1607, 2003. [Online]. Available: http://cancerres.aacrjournals.org/content/63/7/1602
[24] A. Bhattacharjee *et al.*, "Classification of human lung carcinomas by mRNA expression profiling reveals distinct adenocarcinoma subclasses," *Proc. Nat. Acade. Sci. USA*, vol. 98, no. 24, pp. 13 790–13 795, 2001. [Online]. Available: http://www.jstor.org/stable/3057173
[25] S. P. Fodor, "Massively parallel genomics," *Science*, vol. 277, no. 5324, pp. 393–395, 1997.
[26] D. Singh *et al.*, "Gene expression correlates of clinical prostate cancer behavior," *Cancer Cell*, vol. 1, no. 2, pp. 203–209, 2002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1535610802000302

Benjamín Béjar received the electrical engineering degree from the Universitat Politècnica de Catalunya (UPC), Barcelona, Spain, and the Technische Universität Darmstadt (TUD), Darmstadt, Germany, in 2006 under the framework of the double degree exchange program, and the PhD degree in electrical engineering from the Universidad Politécnica de Madrid (UPM), Madrid, Spain, in 2012. Since Jan. 2020 he is a sr. data scientist with the Swiss Data Science Center in Lausanne. He is also an adjunct associate research scientist at the Mathematical Institute for Data Science at Johns Hopkins University. From September 2011 to September 2013 he was a member of the Vision, Dynamics and Learning lab with the Johns Hopkins University (JHU), Baltimore, MD, as part of the MSE program in Biomedical Engineering. He has held research appointments as a visiting PhD student with the Università degli Studi di Udine, Udine, Italy, in 2009 and, at the Hong Kong University of Science and Technology (HKUST), Hong Kong, China, in 2011. In 2012, he was awarded with the 2012 Best Paper Award in Medical Robotics and Computer Assisted Intervention Systems from the Medical Image Computing and Computer Assisted Intervention Society (MICCAI) for his work on surgical activity recognition from video and kinematic data. He served as a postdoctoral researcher and lecturer with the Audiovisual Communications Laboratory (LCAV) at EPFL from October 2013 to October 2017. In October 2017, he joined the Center for Imaging Science with the Johns Hopkins University as an associate research scientist and was later appointed as an assitant research professor with the Department of Biomedical Engineering until 2019. His research interest include signal and image processing, sampling and reconstruction of sparse signals, activity recognition, convex optimization, machine learning, and inverse problems in biomedical data science.

Ivan Dokmanić (Member, IEEE) received the diploma degree in electrical engineering from the University of Zagreb, in 2007 and the doctorate degree in computer and communication science from Ecole Polytechnique Fédérale de Lausanne (EPFL), in 2015. Since 2019, he has been an associate professor at the Department of Mathematics and Computer Science, University of Basel, Switzerland. He is also an adjunct associate professor with the Department of ECE, University of Illinois at Urbana-Champaign, where he was an assistant professor from 2016 to 2020 in the Coordinated Science Laboratory. He has held visiting positions at EPFL (2017, 2018) and ETHZ (2019). From 2015 to 2016 he worked as a postdoc at Institut Langevin and Ecole Normale Supérieure in Paris. During summer 2013, he was with Microsoft Research in Redmond, Washington. Before that he was a teaching assistant with the University of Zagreb, a codec developer for Main Concept AG, and a digital audio effects designer for Little Endian Ltd. His research interests include between data science, physics and signal processing. For his work on room shape reconstruction using sound, he received the Best Student Paper Award at ICASSP 2011; in 2014 he received a Google PhD Fellowship. He is a laureate of the EPFL Outstanding Doctoral Thesis Award and the Google Faculty Research Award. In 2019 the European Research Council (ERC) awarded him a Starting Grant.

René Vidal (Fellow, IEEE) received the BS degree in electrical engineering (valedictorian) from the Pontificia Universidad Católica de Chile, in 1997, and the MS and PhD degrees in electrical engineering and computer science from the University of California at Berkeley, in 2000 and 2003, respectively. He is currently the director of the Mathematical Institute for Data Science (MINDS) and the Hershel L. Seder professor from the Department of Biomedical Engineering, Johns Hopkins University, where he has been since 2004. He is co-author of the book "Generalized Principal Component Analysis" (Springer 2016), co-editor of the book "Dynamical Vision" (Springer 2006), and co-author of more than 250 articles in machine learning, computer vision, signal and image processing, biomedical image analysis, hybrid systems, robotics and control. He is or has been associate editor in chief of the *IEEE Transactions on Pattern Analysis and Machine Intelligence and Computer Vision and Image Understanding*, associate editor or gguest editor of Medical Image Analysis, the *IEEE Transactions on Pattern Analysis and Machine Intelligence*, the *SIAM Journal on Imaging Sciences*, the *Computer Vision and Image Understanding*, the *Journal of Mathematical Imaging and Vision*, the *International Journal on Computer Vision and Signal Processing Magazine*. He has received numerous awards for his work, including the 2016 D'Alembert Faculty Fellowship, the 2012 IAPR J.K. Aggarwal Prize, the 2009 ONR Young Investigator Award, the 2009 Sloan Research Fellowship and the 2005 NSF CAREER Award. He is also fellow of the IAPR, fellow of the AIMBE, and a member of the ACM and SIAM.