FAST PROJECTION ONTO THE $\ell_{\infty,1}$-MIXED NORM BALL USING STEFFENSEN ROOT SEARCH

Gustavo Chau [†] *Brendt Wohlberg*[⋆] *Paul Rodriguez*[†]

[†]Electrical Engineering Department, Pontificia Universidad Católica del Perú, Lima, Peru
[⋆]Theoretical Division, Los Alamos National Laboratory, Los Alamos, NM, USA

ABSTRACT

Mixed norms that promote structured sparsity have broad application in signal processing and machine learning problems. In this work we present a new algorithm for computing the projection onto the $\ell_{\infty,1}$ ball, which has found application in cognitive neuroscience and classification tasks. This algorithm is based on a Steffensen type root search technique, with a number of improvements over prior root search methods for the same problem. First, we theoretically derive an initial guess for the root search algorithm that helps to reduce the number of iterations to be performed. Second, we change the root search method, and through an analysis of the root search function, we construct a pruning strategy that significantly reduces the number of operations. Numerical simulations show that, compared to the state-of-the-art, our algorithm is between 4 and 5 times faster on average, and of up to 14 times faster for very sparse solutions.

Index Terms— Mixed norms, projection, regularization, root search methods

1. INTRODUCTION

Mixed norms are important in modeling group correlations in applications such as genetics [1], electroencephalography [2] and signal processing [3]. In this work we consider mixed norms with non-overlapping groups applied to matrix-form data $\mathbf{A} \in \mathbb{R}^{N\times M}$, where the rows $\mathbf{a}_m \in \mathbb{R}^N$ represent the different groups. Following the notation of [3], the mixed $\ell_{p,q}$-norm of $\mathbf{A}$ is defined as $\|\mathbf{A}\|_{p,q} = (\sum_{m=1}^{M} \|\mathbf{a}_m\|_p^q)^{1/q}$. We will focus on a special case, the $\ell_{\infty,1}$-norm:

$$\|\mathbf{A}\|_{\infty,1} = \sum_{m=1}^{M} \|\mathbf{a}_m\|_\infty \ , \tag{1}$$

where $\|\mathbf{u}\|_\infty = \max_n\{|u_n|\}$ for $\mathbf{u} \in \mathbb{R}^N$.

The main contribution of this work is a computationally efficient algorithm for computing the projection onto the $\ell_{\infty,1}$ ball

$$\text{proj}_{\|\cdot\|_{\infty,1}}(\mathbf{B},\tau) \ := \ \underset{\mathbf{X}}{\text{argmin}} \frac{1}{2}\|\mathbf{X}-\mathbf{B}\|_F^2 \text{ s.t. } \|\mathbf{X}\|_{\infty,1} \leq \tau \ . \tag{2}$$

This $\ell_{\infty,1}$ constraint problem has been applied to image annotation [4], cognitive neuroscience [5] and least absolute shrinkage and selection operator operator (LASSO) regression [6]. Existing algorithms for solving this problem include:

- [7], which can handle general smooth cost function but requires computation of the Hessian.
- [8], which derives an equivalent linear program.
- [4], which uses a projected gradient method.
- [6], which solves the problem by means of a root search technique.

Since [6] is, to the best of our knowledge, the current state-of-the-art for solving the $\ell_{\infty,1}$ constraint problem, we use it as the benchmark for all our comparisons. Our approach for solving (2) is based on the Steffensen root search method [9]. While a general root search based algorithm has previously been proposed for finding the projection onto the $\ell_{p,1}$ ball [6], including the $\ell_{\infty,1}$ case, in this work we propose two modifications that significantly improve the computational performance:

(i) The total number of major iterations for the root search is reduced by applying a simple scheme for choosing a feasible initial solution, devised via a reinterpretation of problem (2).

(ii) An analysis of the search function allows application of a pruning pre-processing stage at each evaluation of the search function, which greatly reduces the total number of projections to be performed.

Our numerical simulations indicate that these improvements reduce the computation time by a factor of 4 – 14 with respect to the state-of-the art methods for solving problem (2).

2. PREVIOUS RELATED METHODS

2.1. Projection onto the $\ell_{p,1}$ ball by root searching

The proximal operator of the $\ell_{p,1}$ norm, defined as

$$\text{prox}_{\|\cdot\|_{q,1}}(\mathbf{B},\lambda) \ := \ \underset{\mathbf{X}}{\text{argmin}} \frac{1}{2}\|\mathbf{X}-\mathbf{B}\|_F^2 + \lambda\|\mathbf{X}\|_{p,1} \ , \tag{3}$$

has a simpler solution than the projection onto the $\ell_{p,1}$-ball (2), since (3) can be computed by solving independent ℓ_p-norm proximity subproblems, i.e. $\text{argmin}_{\mathbf{x}_m} \frac{1}{2}\|\mathbf{x}_m - \mathbf{b}_m\|_F^2 + \lambda\|\mathbf{x}_m\|_p$ [6].

One of the contributions of [6] was to propose a method to take advantage of the separability of (3) in order to solve (2). Let $\mathcal{L}(\mathbf{X},\theta)$ be the Lagrangian of (2), and let θ^* be the dual optimal. As long as $\tau > 0$, (2) satisfies Slater's conditions for strong duality [10], and the primal optimal $\mathbf{X}^*(\theta^*) = \text{argmin}_{\mathbf{X}} \ \mathcal{L}(\mathbf{X},\theta^*)$ is obtained by computing

$$\mathbf{X}^*(\theta^*) \quad = \quad \underset{\mathbf{X}}{\text{argmin}} \frac{1}{2}\|\mathbf{X}-\mathbf{B}\|_F^2 + \theta^*(\|\mathbf{X}\|_{p,1} - \tau) \ . \tag{4}$$

It can be shown [6, Lemma 1] that the scalar function

$$g(\theta) = \|\mathbf{X}(\theta)\|_{p,1} - \tau \ , \tag{5}$$

where $\mathbf{X}(\theta) = \text{prox}_{\|\cdot\|_{p,1}}(\mathbf{B},\theta)$, satisfies the Fourier conditions [11], i.e. in the interval $[0, \ \theta_{\text{max}}]$

(i) $g(0) > 0 > g(\theta\text{max})$

(ii) $g'(\theta) < 0$

(iii) $g''(\theta) \geq 0$.

Additionally, there exists a unique solution for $g(\theta) = 0$, and θ^* coincides with this unique root. Thus, θ^* can be found by using a root finding method.

As for implementation details, [6] recommended a root finding method combining bisection, inverse quadratic interpolation, and the secant method. This root finding based solution for projections onto the $\ell_{p,1}$ ball was extended to the more general $\ell_{p,q}$ case in a follow-up article [12]. Computational performance comparisons indicated [12, Section 3.1] that this extended algorithm was both much more accurate, and twice as fast as that of [4] for projections onto the $\ell_{\infty,1}$ ball.

2.2. Projection onto the ℓ_1-ball

For our solution, we will need to solve the closely related problem of projection onto the ℓ_1-ball, which is defined as

$$\text{proj}_{\|\cdot\|_1}(\mathbf{u}, \tau) = \min_{\mathbf{x}} \frac{1}{2}\|\mathbf{x} - \mathbf{u}\|_2^2 \quad \text{s.t. } \|\mathbf{x}\|_1 \leq \tau \ , \tag{6}$$

where $\mathbf{x}, \mathbf{u} \in \mathbb{R}^N$.

The solution to (6) is given by [13, 14, 15, 12, 16]

$$\mathbf{x}^* = \begin{cases} \mathbf{u} & \text{if } \|\mathbf{u}\|_1 < \tau \\ \text{shrink}(\mathbf{x}, \lambda(\tau)) & \text{if } \|\mathbf{u}\|_1 \geq \tau \ , \end{cases} \tag{7}$$

where $\text{shrink}(\mathbf{x}, \lambda(\tau)) = \text{sign}(\mathbf{x}) \odot \max(|\mathbf{x}| - \lambda(\tau), 0)$, $\odot$ is the element-wise (Hadamard) vector product, and $\lambda(\tau)$ is a shrinkage parameter that depend on τ. Several efficient algorithms exist for finding this shrinkage parameter $\lambda(\tau)$ [17, 18, 19, 14, 20, 13, 21, 16, 22].

2.3. Steffensen's root search method

Steffensen's method is a quasi-Newton root finding algorithm [23] that is useful when an analytical expression of the derivative is not available. The drawback is that it requires two function evaluations, and is therefore usually more costly than Newton's method. Given the function $f(x)$, Steffensen's original iterations consist of the update

$$x_{n+1} := x_n + \frac{x_n}{\delta F(x_n, y_n)} \ , \tag{8}$$

where $\delta F(x_n, y_n) = \frac{f(y_n) - f(x_n)}{y_n - x_n}$ and $y_n = x_n + f(x_n)$.

Steffensen's method tends to exhibit convergence problems if the initial x_0 is too far from the actual root. Therefore, we use the modified version proposed in [23], where $y_n = x_n + \alpha_n |f(x_n)|$. α_n is an adaptive parameter that is recommended to take values that satisfy

$$\text{tol}_c \ll \frac{\text{tol}_u}{2|f(x_n)|} < |\alpha_n| < \frac{\text{tol}_u}{|f(x_n)|} \ , \tag{9}$$

where tol_c is chosen in accordance with the computer precision used in the implementation, and tol_u is a user-defined parameter.

3. PROPOSED METHOD

3.1. Preliminaries

In this section we summarize additional theoretical results from [6] that will be useful in the analysis and derivation of the proposed algorithm.

Lemma 1 (see [6, Lemma 1]). *Let $q \geq 1$ and let q^* be its conjugate exponent satisfying $\frac{1}{q} + \frac{1}{q^*} = 1$. Then, the norm $\|\cdot\|_{p^*,\infty}$ is dual to $\|\cdot\|_{p,1}$.*

Based on the concept of the dual-norm (see Lemma 1) and on Moreau's decomposition [24], [6] proved that the dual problem of

$$\text{prox}_{\|\cdot\|_{\infty,q^*}}(B, \lambda) := \underset{\mathbf{X}}{\text{argmin}} \frac{1}{2}\|\mathbf{X} - \mathbf{B}\|_F^2 + \lambda \cdot \|\mathbf{X}\|_{p^*,\infty} \ , \tag{10}$$

where $\|\mathbf{X}\|_{p^*,\infty} = \max\{\|\mathbf{x}_k\|_{p^*}\}$, is the projection onto the $\ell_{p,1}$ ball $\text{proj}_{\|\cdot\|_{p,1}}(\mathbf{B}, \lambda)$.

While the approach of [6] was solving (10) via (2), which in turn can be effectively solved via a root finding approach, as described in Section 2.1, here we take the reverse route to derive the improvements in the proposed algorithm, as explained in the following section.

3.2. Leveraging $\text{prox}_{\|\cdot\|_{1,\infty}}(\cdot)$, the dual of $\text{proj}_{\|\cdot\|_{\infty,1}}(\cdot)$

Applying the results from Section 3.1 to the specific case of $p = 1$, we observe that the proximal operator of $\ell_{1,\infty}$ is the dual of the projection on the $\ell_{\infty,1}$ ball and vice-versa, then $\mathbf{X}^* = \text{proj}_{\|\cdot\|_{\infty,1}}(\mathbf{B}, \tau)$, can be written as $\mathbf{X}^* = \mathbf{B} - \mathbf{A}^*$, where

$$\mathbf{A}^* = \text{prox}_{\|\cdot\|_{1,\infty}}(\mathbf{B}, \tau) = \min_{\mathbf{A}} \frac{1}{2}\|\mathbf{A} - \mathbf{B}\|_F^2 + \tau \cdot \|\mathbf{A}\|_{1,\infty} \ . \tag{11}$$

Now, if $\mathbf{A}^*$ is known, we can define $\gamma^* = \|\mathbf{A}^*\|_{\infty,1} = \max\{\|a_m^*\|_1\}$, and thus, after simple algebraic manipulation, (11) can be written as

$$\min_{\{a_k\}} \frac{1}{2} \sum_m \|\mathbf{a}_m - \mathbf{b}_m\|_2^2 \quad \text{s.t.} \quad \|\mathbf{a_m}\|_1 \leq \gamma^*, \forall m \ . \tag{12}$$

Clearly, (12) is separable in $\mathbf{a}_m$, with the individual problems corresponding to a projection on the ℓ_1-ball (see Section 2.2). Accordingly, if we devise a method for obtaining the optimal γ^* value, then the solution to (11), and therefore to (2), can be easily calculated. The γ^* value can be found by a root finding method, as described in the following section.

3.3. Search function

As originally proposed in [6], we use $\text{prox}_{\|\cdot\|_{1,\infty}}(\cdot)$ to solve $\text{proj}_{\|\cdot\|_{\infty,1}}(\cdot)$ in (5). Thus, we replace $\mathbf{X}$ by $\mathbf{B} - \mathbf{A}$ and after simple algebraic manipulations, we obtain

$$f(\gamma) \quad = \quad \sum_{m=1}^{M} ||\mathbf{b}_m - \mathbf{a}_m(\gamma)||_\infty - \tau \ , \tag{13}$$

defined for $\gamma \geq 0$. Furthermore, since (13) is equivalent to (5), it also satisfies the Fourier conditions, and thus it has a unique root at γ^*. For a given γ, $\mathbf{a}_m$ is computed using the approach described in (12). As each $\mathbf{a}_m(\gamma)$ corresponds to projections onto the ℓ_1-ball, we apply (7) and obtain

$$\mathbf{a}_m(\gamma) = \begin{cases} \mathbf{b}_m & \text{if } \|\mathbf{b}_m\|_1 < \gamma \\ \text{shrink}(\mathbf{b}_m, \lambda(\gamma)) & \text{if } \|\mathbf{b}_m\|_1 \geq \gamma \ . \end{cases} \tag{14}$$

By substituting (14) into (13), it is observed that only the terms corresponding to the $\|\mathbf{b}_m\|_1 \geq \gamma$ contribute in the sum. Thus, $f(\gamma)$ only depends on the rows of $\mathbf{B}$ that have an ℓ_1-norm greater than γ. Accordingly, at each evaluation of the search function, we can prune the rows of $\mathbf{B}$ that do not fulfill this condition, and only perform the projections specified in (14) on the remaining rows. Our numerical experiments show that this pruning strategy can reduce the computational time by half or more.

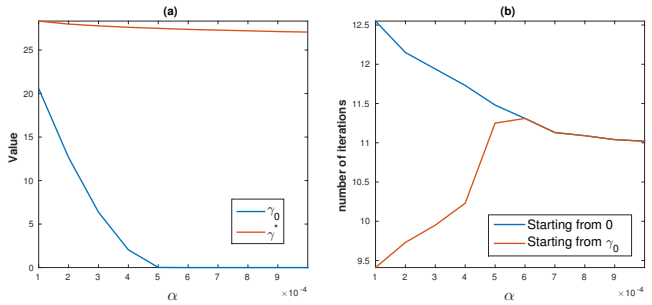

Fig. 1. (a) Value of γ_0 (blue) and γ^* (red) versus α (b) Number of iterations for Steffensen's method to arrive at γ^* starting from 0 (blue) and γ_0 (red) for different α values. See Section 4.1.

3.4. Initial Point

Note that if $\|\mathbf{B}\|_{\infty,1} \leq \tau$ in (2) then the optimal solution is trivial, $\mathbf{X}^* = \mathbf{B}$. From here on, we will consider solely the case where $\|\mathbf{B}\|_{\infty,1} = \sum_{m=1}^{M} \|\mathbf{b}_m\|_\infty > \tau$. Next, we seek to find a point γ_0 such that $f(\gamma_0) > 0$ in (13). Then, as $f(\cdot)$ satisfies the Fourier conditions, we can conclude that $0 \leq \gamma_0 \leq \gamma^*$.

We start by assuming that the ℓ_1-norm of the j^{th} row of the solution $\mathbf{A}^*$ coincides with $\|\mathbf{A}^*\|_{1,\infty}$, i.e., $\max_m\{\|\mathbf{a}_m\|_1\} = \|\mathbf{a}_j\|_1$. Then, via (11), we can find $\mathbf{a}_j$ as:

$$\mathbf{a}_j = \underset{\mathbf{a}}{\text{argmin}} \frac{1}{2}\|\mathbf{a} - \mathbf{b}_j\|_2^2 + \tau\|\mathbf{a}\|_1 = \text{shrink}(\mathbf{b}_j, \tau) \ . \quad (15)$$

If we define $\gamma_0 = \|\text{shrink}(\mathbf{b}_j, \tau)\|_1$, then we get

$$\mathbf{a}_m = \begin{cases} \text{shrink}(\mathbf{b}_j, \tau) & \text{if } m = j \\ \text{proj}_{\|\cdot\|_1}(\mathbf{b}_m, \gamma_0) & \text{if } m \neq j \ . \end{cases} \quad (16)$$

We now proceed to show that $f(\gamma_0) > 0$. We will suppose that $\|\mathbf{b}_j\|_\infty > \tau$ as this is a simple algorithmic check, to be discussed later, that is included in our method. By splitting the sum in (13) for j and $m \neq j$, and using (16), as well as the trivial fact that $\mathbf{b}_j = \text{sign}(\mathbf{b}_j) \odot |\mathbf{b}_j|$, we get

$$f(\gamma_0) = \tau_j(\mathbf{b}_j) + \sum_{m \neq j} ||\mathbf{b}_m - \text{proj}_{\|\cdot\|_1}(\mathbf{b}_m, \gamma_0)||_\infty - \tau \ , \quad (17)$$

where $\tau_j(\mathbf{b}_j) = \||\mathbf{b}_j| - \max(|\mathbf{b}_j| - \tau, 0)\|_\infty$. Now, we turn to the analysis of $\tau_j(\mathbf{b}_j)$. As all the components involved in $\tau_j(\mathbf{b}_j)$ are positive, we can write the norm as the maximum of all the components of the vector. For each component, we have:

$$|b_j^{(i)}| - \max(|b_j^{(i)}| - \tau, 0) = \begin{cases} \tau & \text{if } |b_j^{(i)}| > \tau \\ |b_j^{(i)}| & \text{if } |b_j^{(i)}| \leq \tau \ . \end{cases} \quad (18)$$

Since we assumed that $\|\mathbf{b}_j\|_\infty > \tau$, then at least one of the components of $\mathbf{b}_j$ must fulfill the first condition in (18). For at least this component, we have $|b_j^{(i)}| - \max(|b_j^{(i)}| - \tau, 0) = \tau$ and all the other components are less than or equal to τ. Accordingly, $\||\mathbf{b}_j| - \max(|\mathbf{b}_j| - \tau, 0)\|_\infty = \tau$. Replacing this in (17), we obtain

$$f(\gamma_0) \quad = \quad \sum_{m \neq j} ||\mathbf{b}_m - \text{proj}_{\|\cdot\|_1}(\mathbf{b}_m, \gamma_0)||_\infty > 0 \ . \quad (19)$$

Thus, $f(\gamma_0) > 0$ and, instead of starting the root search from 0, we can start from γ_0, which is a better initial guess of γ^*. *A priori*, we do not know which j is closer (or corresponds) to the real maximum. In order to obtain the initial point γ_0, we solve (15) for every row and then take among these solutions the one with the maximum ℓ_1-norm.

3.5. Proposed method

The full proposed method is presented in Algorithm 1. Note that if $\|\mathbf{B}\|_{\infty,\infty} < \tau$, line 3 does not need to be evaluated and γ is assigned an initial value of 0. Likewise, the shrinkage operation is performed solely for the rows whose ℓ_∞-norm is greater than τ.

Algorithm 1: Proposed Method

Input: matrix $\mathbf{B}$, τ, maxIter, tolerance
1 **if** $\|\mathbf{B}\|_{\infty,1} \leq \tau$ **then**
2 return $\mathbf{B}$
3 Compute $\alpha_k = \|\text{shrink}(b_k, \tau)\|_1$ for each row of B. Define $\gamma = \max_k(\alpha_k)$
4 **for** $k = 1 : \text{maxIter}$ **do**
5 Prune the rows of B that have ℓ_1-norm less than γ
6 Obtain $f(\gamma)$ as defined in (13)
7 **if** $|f(\gamma)| < \text{tolerance}$ **then**
8 break
9 Update γ using Steffensen method
10 **end**
11 Obtain $\mathbf{A}$ as defined in (12) using the obtained γ
12 Return $\mathbf{B} - \mathbf{A}$

Table 1. Computational results comparing the effect of the initial point γ_0. The percent change from the zero-start case is shown in parenthesis for the γ_0 case. See Section 4.1.

	Starting at zero		**Starting at γ_0**			
α/ sparsity(%)	**num iter**	**time(s)**	**num iter**		**time(s)**	
0.0001/ 1.02	12.6	1.1	9.4	(-25%)	0.63	(-43%)
0.0002/ 1.92	12.2	1.1	9.7	(-20%)	0.71	(-35%)
0.0003/ 2.68	11.9	1.1	10.0	(-17%)	0.77	(-29%)
0.0004/ 3.40	11.7	1.1	10.2	(-13%)	0.86	(-20%)
0.0005/ 4.17	11.5	1.1	11.3	(-2%)	1.11	(4%)
0.0006/ 4.94	11.3	1.1	11.3	(0%)	1.06	(0%)
0.0007/ 5.53	11.1	1.0	11.1	(0%)	1.05	(0%)
0.0008/ 6.28	11.1	1.1	11.1	(0%)	1.05	(0%)
0.0009/ 6.89	11.0	1.0	11.0	(0%)	1.04	(0%)
0.0010/ 7.53	11.0	1.1	11.0	(0%)	1.06	(0%)

4. RESULTS

All tests presented below were computed using single-threaded Matlab code running on an Intel i7-4770K CPU (8 cores, 3.50 GHz, 32GB RAM). Matrix $\mathbf{B}$ was generated using a uniform distribution $[-0.5, 0.5]$, and τ, the constraint used in (1), was taken such that $\tau = \alpha\|\mathbf{B}\|_{\infty,1}$, where α is a small constant. Specific sizes of $\mathbf{B}$ and values of α are mentioned below. Our Matlab code [25] can be used to reproduce our experimental results.

4.1. Impact of initial point

In order to study the impact of the initial point γ_0 on the performance of the algorithm, we constructed 100 different realizations of a 2000×100 $\mathbf{B}$ matrix, considering[1] $\alpha \in [10^{-4}, 10^{-3}]$. For each value of τ, the values for the initial point γ_0 and the optimal value γ^* were averaged across the 100 realizations. These average values for each τ are shown in Figure 1(a). It is observed that, at low α values, γ_0 is very close to the optimal value, but it goes rapidly to zero as τ increases. On the other hand, Figure 1(b) shows a comparison of the average number of iterations that the proposed method needs for arriving to the optimal value starting from either zero or

Table 2. Results for simulations with matrices of different size and the three tested methods. Error (Err.), number of iterations (N.I.) and running times are shown for each of them. Speedup with respect to Sra is shown for Proposed and Proposed + pruning.

		Sra [6]			**Proposed**				**Proposed + pruning**			
Matrix Size	α **/ sparsity(%)**	**Err.**	**N.I.**	**Time(s)**	**Err.**	**N.I.**	**Time(s)**	**Speedup**	**Err.**	**N.I.**	**Time(s)**	**Speedup**
2 000 x 100	0.0001/ 1.02	3.4e-11	9.6	4.04	2.6e-12	9.4	0.64	**6.31**	2.6e-12	9.4	0.27	**14.96**
	0.0005/ 4.13	1.5e-10	13.2	4.20	1.4e-12	11.3	1.12	**3.75**	1.4e-12	11.3	0.84	**4.98**
	0.0010/ 7.51	5.4e-10	14.5	4.41	1.3e-12	11.0	1.07	**4.12**	1.3e-12	11.0	0.82	**5.36**
5 000 x 200	0.0001/ 1.37	1.7e-10	17.0	13.99	1.6e-12	10.6	2.35	**5.95**	1.6e-12	10.6	1.27	**10.99**
	0.0005/ 5.59	3.4e-10	11.9	13.67	8.0e-12	12.0	3.32	**4.12**	8.0e-12	12.0	2.44	**5.61**
	0.0010/ 10.03	7.8e-10	17.9	14.10	6.2e-13	11.1	3.23	**4.37**	6.0e-12	11.1	2.57	**5.50**
10 000 x 300	0.0001/ 1.62	4.0e-10	11.4	27.74	9.4e-14	13.0	7.97	**3.48**	9.8e-14	13.0	5.71	**4.86**
	0.0005/ 6.6	2.1E-09	14.5	29.03	6.9e-14	12.0	7.99	**3.63**	7.0e-14	12.0	5.64	**5.15**
	0.0010/ 11.88	3.3E-09	15.7	37.09	2.8e-12	11.4	7.81	**4.75**	2.8e-12	11.4	5.79	**6.41**

γ_0. The number of iterations and computational time for the different τ values, for the proposed method without pruning, along with the improvements provided by using γ_0, are listed in Table 1

4.2. Description of the comparisons

We compare our proposed method without and with the pruning step (denoted as "Proposed" and "Proposed with pruning", respectively) against [6], denoted as Sra. Unfortunately, we could not find a public implementation of [6], and thus, we coded our own Matlab version using the `fzero` function as root search method as suggested in [6].

We chose $\text{tol}_c = 10^{-12}$ and $\text{tol}_u = 10^{-8}$ in (9) for the Steffensen root search method. The projections onto the ℓ_1-ball, needed for the evaluation of the search function in [6] and in our algorithm, are implemented using Michelot's algorithm [26]. This algorithm was chosen since it can be implemented efficiently [22, Section 3.2.2] and, for small size projections, we have empirically observed that it has better computational performance than the alternatives mentioned in Section 2.2.

We simulated three different sizes for the matrix $\mathbf{B}$, namely[1] 2000×100, 5000×200 and 10000×300. 100 realizations of $\mathbf{B}$ were taken. We also considered[1] $\alpha \in \{10^{-4}, 5 \times 10^{-4}, 10^{-3}\}$, to experimentally obtain approximate sparsity percentages (percentage of non-zero rows) of 1, 5 and 10%, respectively.

For all simulations, we recorded the error, measured as the constraint violation $|\|\mathbf{X}\|_{\infty,1} - \tau|$, the number of iterations, and the total time until convergence. We additionally computed a sparsity value as the percentage of non-zero rows. As all methods arrive to the same value of sparsity, only a single value is shown for each α. The results averaged over the 100 realizations for the different matrix dimensions are shown in Table 2.

4.3. Discussion of results

As can be observed from the results of Section 4.1, the initial point γ_0 has impact only at low α (and therefore τ) values. This is easily explainable, as at high τ values the solution of (15) is zero. Accordingly, as suggested in Section 3.5, it is better to first evaluate the conditions on $\|\mathbf{B}\|_{\infty,\infty}$ and $\|\mathbf{b}_i\|_\infty$ to avoid unnecessary shrinkage operations, as these comparisons do not incur a great computational cost. When the initial point is different from zero, we see that the number of iterations and the computational time are slightly reduced. For our tests, when τ_0 is not zero, an average reductions of 2.2 iterations (-15%) and of 24% in time are achieved.

[1]These sizes and sparsity values are typical for known applications of (1); see [4, 5, 6]. Results for larger values of α can be obtained with our source code [25].

As indicated by the results of Section 4.2, the proposed method tends to require fewer iterations than Sra's method, although this is not directly comparable as the stopping criteria of the `fzero` function is different from the one we are using in Algorithm 1, and because our proposed algorithms are obtaining smaller errors. Nevertheless, our proposed algorithm achieves a lower constraint error in less total time, with speedups of a factor of around 3 – 4 for "Proposed" and more than 5.5 for "Proposed+pruning". These speedups tended to be higher at low τ values, as expected from the initial point analysis, and do not seem to be related to the size of the matrix. As expected, the pruning step does not change the number of iterations or the error significantly (the differences are due to finite precision arithmetic) but reduces the time of root search by approximately 33%. Overall, the proposed method with pruning presents speedups with respect to Sra's method ranging from 5 – 6 on average and going up to 10 – 14 for cases of higher sparsity in the solutions.

5. CONCLUSION

We have presented a new algorithm for projection onto the $\ell_{\infty,1}$-norm ball that exploits the particular structure of this problem to improve over previous methods. By analyzing Moreau's decomposition and the dual norm of the original problem, we derived an initial guess that reduces the number of iterations for low constraint values. In addition, the analysis of the search function allows us to use a pruning strategy that considerably reduces the computational cost.

Empirically we have shown that the proposed method can provide speedups of around 5 – 6 times in most cases, and of up to 10 – 14 times in favorable sparsity conditions, and that the accuracy of our solution is at least one order of magnitude better than the state-of-the-art.

6. REFERENCES

[1] L. Yuan, J. Liu, and J. Ye, "Efficient methods for overlapping group lasso," in *Advances in Neural Information Processing Systems*, 2011, pp. 352–360.

[2] A. Gramfort, M. Kowalski, and M. Hämäläinen, "Mixed-norm estimates for the M/EEG inverse problem using accelerated gradient methods," *Physics in medicine and biology*, vol. 57, no. 7, p. 1937, 2012.

[3] M. Kowalski, "Sparse regression using mixed norms," *Applied and Computational Harmonic Analysis*, vol. 27, no. 3, pp. 303–324, 2009.

[4] A. Quattoni, X. Carreras, M. Collins, and T. Darrell, "An efficient projection for $\ell_{1,\infty}$ regularization," in *Proceedings of the*

26th Annual International Conference on Machine Learning. ACM, 2009, pp. 857–864.

[5] H. Liu, M. Palatucci, and J. Zhang, "Blockwise coordinate descent procedures for the multi-task lasso, with applications to neural semantic basis discovery," in *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 2009, pp. 649–656.

[6] S. Sra, "Fast projections onto $\ell_{1,q}$-norm balls for grouped feature selection," *Machine learning and knowledge discovery in databases*, pp. 305–317, 2011.

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

[8] A. Quattoni, M. Collins, and T. Darrell, "Transfer learning for image classification with sparse prototype representations," in *IEEE Conference on Computer Vision and Pattern Recognition, 2008*, 2008, pp. 1–8.

[9] L. Johnson and D. Scholz, "On Steffensen's method," *SIAM Journal on Numerical Analysis*, vol. 5, no. 2, pp. 296–302, 1968.

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

[11] A. G. Akritas, "On the Budan-Fourier controversy," *ACM SIGSAM Bulletin*, vol. 15, no. 1, pp. 8–10, 1981.

[12] S. Sra, "Fast projections onto mixed-norm balls with applications," *Data Mining and Knowledge Discovery*, vol. 25, no. 2, pp. 358–377, Sep 2012.

[13] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra, "Efficient projections onto the ℓ_1-ball for learning in high dimensions," in *ICML*, 2008, pp. 272–279.

[14] J. Liu and J. Ye, "Efficient Euclidean projections in linear time," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2009, pp. 657–664.

[15] S. Sra, "Generalized proximity and projection with norms and mixed-norms," Max Planck Institute for Biological Cybernetics, Tübingen, Germany, Tech. Rep. 192, may 2010.

[16] L. Condat, "Fast projection onto the simplex and the ℓ_1 ball," *Mathematical Programming*, vol. 158, no. 1-2, pp. 575–585, 2016.

[17] M. Held, P. Wolfe, and H. Crowder, "Validation of subgradient optimization." *Math. Program.*, vol. 6, no. 1, pp. 62–88, 1974.

[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] E. van den Berg and M. Friedlander, "Probing the Pareto frontier for basis pursuit solutions," *SIAM Journal on Scientific Computing*, vol. 31, no. 2, pp. 890–912, 2009.

[20] K. Kiwiel, "Breakpoint searching algorithms for the continuous quadratic knapsack problem," *Mathematical Programming*, vol. 112, no. 2, pp. 473–491, 2008.

[21] P. Gong, K. Gai, and C. Zhang, "Efficient Euclidean projections via piecewise root finding and its application in gradient projection," *Neurocomputing*, vol. 74, no. 17, pp. 2754–2766, 2011.

[22] P. Rodríguez, "An accelerated Newton's method for projections onto the $\ell_l 1$-ball," in *IEEE International Workshop on Machine Learning for Signal Processing (MSLP)*, 2017.

[23] S. Amat, S. Busquier, Á. Magreñán, and L. Orcos, "An overview on Steffensen-type methods," in *Advances in Iterative Methods for Nonlinear Equations*. Springer, 2016, pp. 5–21.

[24] N. Parikh, S. Boyd *et al.*, "Proximal algorithms," *Foundations and Trends in Optimization*, vol. 1, no. 3, pp. 127–239, 2014.

[25] G. Chau and P. Rodriguez, "Simulations for fast projection onto the $\ell_{1,\infty}$-mixed norm ball using Steffensen root search," https://goo.gl/TngGrc.

[26] C. Michelot, "A finite algorithm for finding the projection of a point onto the canonical simplex of $\mathbb{R}^n$," *Journal of Optimization Theory and Applications*, vol. 50, no. 1, pp. 195–200, 1986.