2015 IEEE 54th Annual Conference on Decision and Control (CDC)
December 15-18, 2015. Osaka, Japan

Projected sub-gradient with ℓ_1 or simplex constraints via isotonic regression

Jérôme Thai[1] Cathy Wu[1] Alexey Pozdnukhov[2] Alexandre Bayen[1,2]

Abstract— **We consider two classic problems in convex optimization: 1) minimizing a convex objective over the nonnegative orthant of the ℓ_1-ball and 2) minimizing a convex objective over the probability simplex. We propose an efficient and simple equality constraint elimination technique which converts the ℓ_1 and simplex constraints into order constraints. We formulate the projection onto the feasible set as an isotonic regression problem, which can be solved exactly in $O(n)$ time via the Pool Adjacent Violators Algorithm (PAVA), where n is the dimension of the space. We design a C++ implementation of PAVA up to 25,000 times faster than `scikit-learn`. Our PAVA-based projection step enables the design of efficient projected subgradient methods which compare well against projected algorithms using direct projections onto the ℓ_1-ball and onto the simplex, with projection in $O(n\log(n))$ exact time and $O(n)$ expected time. Interestingly, our technique is particularly well adapted to learning from sparse, skewed, or aggregated data, by decreasing the cross-correlations between data points.**

I. Introduction

We study the following convex optimization problems:

$$\min_{\mathbf{x}} f(\mathbf{x}) \quad \text{s.t.} \quad \|\mathbf{x}\|_1 \leq \lambda, \quad \mathbf{x} \succeq 0 \tag{1}$$

$$\min_{\mathbf{x}} f(\mathbf{x}) \quad \text{s.t.} \quad \mathbf{1}^T\mathbf{x} = \lambda, \quad \mathbf{x} \succeq 0 \tag{2}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a convex function, $\lambda, \lambda_1, \cdots, \lambda_s$ positive constants, $\|\mathbf{x}\|_1 = \sum_{i=1}^n |x_i|$ is the ℓ_1-norm, $\mathbf{1}^T\mathbf{x} = \sum_{i=1}^n x_i$, and $\mathbf{x} \succeq 0$ describes the nonnegative orthant $x_i \geq 0, i \in \{1, \cdots, n\}$. In the third problem, the vector $\mathbf{x}$ is decomposed into s subvectors $\mathbf{x}^1, \cdots, \mathbf{x}^s$ such that each block $\mathbf{x}^k$ is of dimension n_k (with $\sum_k n_k = n$) and subject to equality constraints $\mathbf{1}^T\mathbf{x}^k = \sum_{i=1}^{n_k} x_i^k = \lambda_k$ for $k \in \{1, \cdots, s\}$.

In our setting, $\|\mathbf{x}\|_1 = \sum_{i=1}^n x_i = \mathbf{1}^T\mathbf{x}$ since the entries of $\mathbf{x}$ are restricted to be nonnegative, hence we will use $\mathbf{1}^T\mathbf{x}$ instead of the norm notation for the remainder of the article. We focus on projected subgradient methods, and projecting onto the ℓ_1-ball is equivalent to projecting the absolute value of x onto the nonnegative orthant of the ℓ_1-ball [8], thus for brevity, in the remainder of the article, we will refer to the latter as just projecting on the ℓ_1-ball.

Problems with a cardinality penalty term are common in machine learning. They are often cast into a convex optimization formulation in which the ℓ_1 norm is a proxy for penalizing cardinality. Such problems have many applications, see [8], [17] and references therein. An equivalent approach to cast the ℓ_1 regularized problem is to impose a ℓ_1 constraint, such as in problem (1). A mathematically related task in machine learning is the estimation on the probability simplex, see Pilanci et al. [17] and references therein. Applications include measure recovery [17], portfolio optimization [13], traffic assignment [16], route flow estimation in transportation networks [19]. Duchi et al. [8] focuses on the ℓ_1 constrained problem (1) and designs efficient projected subgradient methods with projection in $O(n\log n)$ in exact time or $O(n)$ expected time.

In this paper, we present an efficient equality constraint elimination technique [5, §4.1.3] on the simplex constrained problem (2) which reduces the constraints to order constraints. Hence isotonic regression techniques, see [18], can be used and by extension the ℓ_1 constrained problem (1). Isotonic regression can be solved in linear time by the *Pool Adjacent Violators Algorithm* (PAVA) [4, §3], and has been successfully applied to Ad Click Prediction [14], [10]. We also design a C++ implementation of PAVA that is approximately 24,000 times faster than the PAVA from the widely-used Python library `scikit-learn`, and 5-10 times faster than the state-of-the-art. Thus, the proposed parametrization $\mathbf{x} = \mathbf{N}\mathbf{z} + \mathbf{e}$ allows for efficient gradient projection in linear time using PAVA and compares well against the projection on the ℓ_1-ball, which has $O(n)$ expected time complexity [8].

We compare the rates of converence of different projection subgradient descent methods: the Barzilai-Borwein method [2] and the L-BFGS (see [15]), augmented with a backtracking line search [5, §9.2] on problems (1) and (2). We demonstrate experimentally that our method performs well on learning from sparse data points sampled from skewed distributions. We note that many models assume that data are symmetric about the mean, *e.g.* sampled from a normal distribution. In reality, data points may not be perfectly symmetric and there is active research on skewness [12].

II. Problem setting and existing techniques

Our analysis is motivated by devising efficient projected subgradient methods, see, *e.g.* Bertsekas [3]. We present how (1) can be solved using such methods, see Duchi et al. [9]. Throughout the paper, we will denote the feasible set of (1), which is the intersection of the ℓ_1-ball and $\mathbb{R}^n_+$:

$$B_+^\lambda = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{1}^T\mathbf{x} \leq \lambda, \mathbf{x} \succeq 0\} \tag{3}$$

A. Gradient projection onto the ℓ_1-ball

Projected subgradient methods minimize the objective f subject to $\mathbf{x} \in B_+^\lambda$ by computing the sequence $\mathbf{x}_0, \mathbf{x}_1, \cdots$ with

$$\mathbf{x}_{t+1} = \Pi_{B_+^\lambda}\Big(\mathbf{x}_t - \gamma_t \nabla f(\mathbf{x}_t)\Big), \quad t = 0, 1, 2, \cdots \tag{4}$$

[1]Department of Electrical Engineering and Computer Sciences, University of California at Berkeley.
[2]Department of Civil and Environmental Engineering, University of California at Berkeley.
contact: `jerome.thai@berkeley.edu`

where $\nabla f(\mathbf{x}_t)$ is the (sub)gradient of f at $\mathbf{x}_t$, γ_t is a positive step size in the direction of negative gradient, and $\Pi_{B_+^\lambda}(\mathbf{w})$ is the Euclidean projection of $\mathbf{w}$ onto B_+^λ, *i.e.* the solution of

$$\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{w}-\mathbf{x}\|_2^2 \quad \text{s.t.} \quad \mathbf{1}^T\mathbf{x} \leq \lambda,\, \mathbf{x} \succeq 0 \tag{5}$$

The above problem (5) can be solved by first projecting $\mathbf{w}$ onto $\mathbb{R}_+^n$, *i.e.* setting all the negative entries to zero. If the resulting vector $\mathbf{w}_+ := (\max(w_i,0))_{i=1,\cdots,n}$ is such that $\mathbf{1}^T\mathbf{w}_+ \leq \lambda$, then the optimal solution of (5) is $\mathbf{w}_+$. If we have $\mathbf{1}^T\mathbf{w}_+ > \lambda$, the optimal solution must be on the boundary of the ℓ_1-ball and thus the inequality constraint $\mathbf{1}^T\mathbf{x} \leq \lambda$ can be replaced with the equality constraint $\mathbf{1}^T\mathbf{x} = \lambda$:

$$\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{w}-\mathbf{x}\|_2^2 \quad \text{s.t.} \quad \mathbf{1}^T\mathbf{x} = \lambda,\, \mathbf{x} \succeq 0 \tag{6}$$

A unique solution to problem (6) exists and can be obtained by shifting all the entries in $\mathbf{w}$ by the same amount $-\theta$ while keeping all the entries nonnegative, *i.e.* finding $\theta \in \mathbb{R}$ such that the vector$(\mathbf{w}-\theta\mathbf{1})_+ = (\max(w_i-\theta,0))_{i=1,\cdots,n}$ is feasible $\mathbf{1}^T(\mathbf{w}-\theta\mathbf{1})_+ = \lambda$. Computing the optimal threshold θ requires sorting $\mathbf{w}$ in decreasing order $w_{(1)}, w_{(2)}, \cdots, w_{(n)}$ and finding the pivot $\rho \in [n]$ and then θ such that $w_{(i)} - \theta > 0$ for $1 \leq i \leq \rho$ and $w_{(i)} - \theta \leq 0$ otherwise. The sorting step is the most expensive one and requires $O(n \log n)$ time (using *e.g.* quicksort). An improvement [9] consists of finding the pivot element ρ in step 2 in expected linear time without sorting $\mathbf{w}$. It is based on a modification of the randomized median finding algorithm of Cormen et al. [6].

We note that problems on the simplex (2) are similar with a projection step that is directly given by (6). In the remainder of the paper, we will denote S_λ^n the simplex of mass λ:

$$S_\lambda^n = \{\mathbf{x} \in \mathbb{R}^n \,|\, \mathbf{1}^T\mathbf{x} = \lambda,\, \mathbf{x} \succeq 0\} \tag{7}$$

III. Our constraint elimination technique

In this section, we present one of our main contributions: a simple and efficient equality constraint elimination technique that converts the simplex constraints into order constraints. Our method also eliminates the equality constraint in the projection step of ℓ_1-constrained problems.

A. Reduction to an order constraint

Using linear algebra [5, §4.1.3], we eliminate the equality constraint $\mathbf{1}^T\mathbf{w} = \lambda$ in (2) by constructing a feasible direction $\mathbf{e} \in S_\lambda^n$ and a matrix $\mathbf{N} \in \mathbb{R}^{n\times(n-1)}$ whose range is the orthogonal complement of the vector $\mathbf{1} \in \mathbb{R}^n$, *i.e.* $\{t\mathbf{1} \,|\, t \in \mathbb{R}\}^\perp$. With such components $\mathbf{e}$ and $\mathbf{N}$, we have $\{\mathbf{x} \in \mathbb{R}^n \,|\, \mathbf{1}^T\mathbf{w} = \lambda\} = \{\mathbf{e}+\mathbf{N}\mathbf{z} \,|\, \mathbf{z} \in \mathbb{R}^{n-1}\}$, and substituting $\mathbf{x} = \mathbf{e}+\mathbf{N}\mathbf{z}$ in (2) gives:

$$\min_{\mathbf{z}} f(\mathbf{e}+\mathbf{N}\mathbf{z}) \quad \text{s.t.} \quad \mathbf{e}+\mathbf{N}\mathbf{z} \succeq 0 \tag{8}$$

Proposition 1: *There exists an affine transformation* $\mathbf{x} = \mathbf{e}+\mathbf{N}\mathbf{z}$ *(8) such that the simplex-constrained problem (2) is equivalent to a minimization problem with order constraint:*

$$\min_{\mathbf{z}} f(\mathbf{e}+\mathbf{N}\mathbf{z}) \quad \textit{s.t.} \quad 0 \leq z_1 \leq \cdots \leq z_{n-1} \leq \lambda \tag{9}$$

Proof: Vectors of the form $[0,\cdots,1,-1,\cdots,0]^T$ are orthogonal to $\mathbf{1} \in \mathbb{R}^n$, hence we choose $\mathbf{N}$ and $\mathbf{e} \in \mathbb{R}^n$ such that:

$$\mathbf{N} = \begin{bmatrix} 1 & & \\ -1 & 1 & \\ & -1 & \ddots \\ & & \ddots \end{bmatrix} \in \mathbb{R}^{n\times(n-1)}; \quad \mathbf{e} =: [0,\cdots,0,\lambda]^T \in \mathbb{R}^n \tag{10}$$

where the columns of $\mathbf{N}$ form a basis of $\{t\mathbf{1} \,|\, t \in \mathbb{R}\}^\perp$. With the above construction, the entries of $\mathbf{x} = \mathbf{e}+\mathbf{N}\mathbf{z}$ are

$$\begin{aligned} x_1 &= z_1 \\ x_i &= z_i - z_{i-1} \quad i = 2,\cdots,n-1 \\ x_n &= \lambda - z_{n-1} \end{aligned} \tag{11}$$

which results in a simplification of the constraint $\mathbf{e}+\mathbf{N}\mathbf{z} \succeq 0$ into order constraints $0 \leq z_1 \leq \cdots \leq z_{n-1} \leq \lambda$. □

This also applies to ℓ_1-constrained problems:

Proposition 2: *There exists an affine transformation* $\mathbf{x} = \mathbf{N}\mathbf{z}$ *such that the ℓ_1-constrained problem (1) is equivalent to a minimization problem with order constraint of the form:*

$$\min_{\mathbf{z}} f(\mathbf{N}\mathbf{z}) \quad \textit{s.t.} \quad 0 \leq z_1 \leq \cdots \leq z_n \leq \lambda \tag{12}$$

Proof: We note that the variables $x_1,\cdots,x_n$ given by (11) form a telescoping sum, yielding $x_1+\cdots+x_n = \lambda$, which is the desired equality constraint in the simplex. For the ℓ_1-ball restricted to $\mathbb{R}_+^n$, we want $x_1+\cdots+x_n \leq \lambda$, which is obtained by replacing the last equality in (11) by the inequality $x_n \leq \lambda - z_{n-1}$. If we pose $z_n := x_n + z_{n-1}$, then we have $z_n \leq \lambda$ and $x_n = z_n - z_{n-1} \geq 0$, which set the constraints on the added variable z_n to $z_{n-1} \leq z_n \leq \lambda$. Hence, for ℓ_1-constrained problems, the affine transformation is $\mathbf{x} = \mathbf{N}\mathbf{z}$ with

$$\mathbf{N} = \begin{bmatrix} 1 & & & \\ -1 & \ddots & & \\ & \ddots & 1 & \\ & & -1 & 1 \end{bmatrix} \in \mathbb{R}^{n\times n} \tag{13}$$

$$\begin{aligned} x_1 &= z_1 \\ x_i &= z_i - z_{i-1} \quad i = 2,\cdots,n \end{aligned} \qquad \square \tag{14}$$

B. Geometric and statistical interpretations

We observe that the columns of $\mathbf{N}$ in (13) do not form an orthogonal basis for the vector $\mathbf{z}$, the scalar product for two consecutive vectors being -1. Hence, the projection in the $\mathbf{z}$-basis is not equivalent to the one in the $\mathbf{x}$-basis.

We also note that $\mathbf{N}$ in (10) or (13) is full rank, hence our transformation does not affect the strong or weak convexity of the objective function f. However, as we will see later, the condition number of the sublevel sets of f is affected, see [5, §9.1]. Finally, we give a statistical interpretation: inverting equations (14), the variables z_i are $z_i = \sum_{j=1}^i x_j$, hence $\mathbf{z}$ is the cumulative sum of the entries of $\mathbf{x}$. If $\mathbf{x}$ is constrained in the probability simplex, then $\mathbf{z}$ describes the cumulative density function associated to $\mathbf{x}$.

IV. Projection as an isotonic regression problem

In this section, we focus on problems of the form (12), the analysis of (9) being the same. The sequence of points generated by projected gradient methods applied to (12) is:

$$\mathbf{z}_{t+1} = \Pi^{\text{iso}}_{[0,\lambda]}\Big(\mathbf{z}_t - \gamma_t \mathbf{N}^T \nabla f(\mathbf{N}\mathbf{z}_t)\Big) \tag{15}$$

where $\Pi^{\text{iso}}_{[0,\lambda]}(\mathbf{y})$ is the Euclidean projection of $\mathbf{y}$ onto the order constraints:

$$\min_{\mathbf{x}} \sum_{i=1}^{n} (y_i - x_i)^2 \quad \text{s.t.} \quad 0 \le x_1 \le x_2 \le \cdots \le x_n \le \lambda \tag{16}$$

Without the lower and upper bounds, we are left with an isotonic regression problem, denoted $\Pi^{\text{iso}}_{\mathbb{R}}(\mathbf{y})$

$$\min_{\mathbf{x}} \sum_{i=1}^{n} (y_i - x_i)^2 \quad \text{s.t.} \quad x_1 \le x_2 \le \cdots \le x_n \tag{17}$$

Problem (17) has a unique solution which can be obtained using the Pool Adjacent Violators Algorithm (PAVA) [1]. PAVA starts with y_1 on the left and move to the right until it encounters the first violation $y_i > y_{i+1}$. Then it replaces this pair by their average, and back-average to the left as needed, to get monotonicity. Then it continues this process to the right, until finally it reaches y_n. It has been shown that PAVA terminates in $O(n)$ time [11]. Hence we would like to reduce the box-constrained problem (16) to (17) so that we can use PAVA to perform our gradient projection.

A. Isotonic regression with box constraints

Throughout our analysis, we denote $\Pi^{\text{iso}}_{[a,b]}(\mathbf{y}_{k\to l})$ the problem of fitting an increasing subsequence $a \le x_k, x_{k+1}, \cdots, x_l \le b$ to $y_k, y_{k+1}, \cdots, y_l$:

$$\min \sum_{i=k}^{l} (y_i - x_i)^2 \quad \text{s.t.} \quad a \le x_k \le x_{k+1} \le \cdots \le x_l \le b \tag{18}$$

Hence we would like to reduce the box-constrained problem $\Pi^{\text{iso}}_{[0,\lambda]}(\mathbf{y})$ to $\Pi^{\text{iso}}_{\mathbb{R}}(\mathbf{y})$. We provide a lemma and the main result:

Lemma 1. *Given $\mathbf{x}^{iso}$ the solution to (17), if there exists k such that $x^{iso}_k < x^{iso}_{k+1}$ then (17) reduces to the two subproblems $\Pi^{iso}_{\mathbb{R}}(\mathbf{y}_{1\to k})$ and $\Pi^{iso}_{\mathbb{R}}(\mathbf{y}_{k+1\to n})$*

$$\begin{array}{l} \min \sum_{i=1}^{k} (y_i - x_i)^2 \quad s.t. \quad x_1 \le \cdots \le x_k \\ \min \sum_{i=k+1}^{n} (y_i - x_i)^2 \quad s.t. \quad x_{k+1} \le \cdots \le x_n \end{array} \tag{19}$$

such that $[x^{iso}_1, \cdots, x^{iso}_k]$ is the solution to the former and $[x^{iso}_{k+1}, \cdots, x^{iso}_n]$ is the solution to the latter. The same result holds for (16) and $\mathbf{x}^\star$, with resulting subproblems $\Pi^{iso}_{[0,+\infty)}(\mathbf{y}_{1\to k})$ and $\Pi^{iso}_{(-\infty,\lambda]}(\mathbf{y}_{k+1\to n})$.

Proof: Since the constraint $x_k \le x_{k+1}$ is not active at $\mathbf{x}^{\text{iso}}$, it may be removed without altering the solution. Then the resulting program is separable into the two programs in (19) with respective solutions $[x^{\text{iso}}_1, \cdots, x^{\text{iso}}_k]$ and $[x^{\text{iso}}_{k+1}, \cdots, x^{\text{iso}}_n]$.□

Proposition 3: *The solution $\mathbf{x}^\star$ to (16) is the Euclidian projection of the solution $\mathbf{x}^{iso}$ to (17) onto $[0,\lambda]^n$,* i.e. *$x^\star_i = x^{iso}_i$ if $0 \le x^{iso}_i \le \lambda$, $x^\star_i = 0$ if $x^{iso} < 0$, and $x^\star_i = \lambda$ if $x^{iso} > \lambda$.*

Proof: We only prove for a box of the form $[0, +\infty)$, since the discussion extends straightforwardly to the case $[0,\lambda]$. For correctness, we add the component $x^\star_0 = 0$ to $\mathbf{x}^\star$, $\mathbf{x}^\star$ being the solution to (16).

Special case: $[x^{\text{iso}}_i \le 0, \forall i]$. Suppose $\exists k \in \{1, \cdots, n\}, x^\star_k > 0$. We choose k the smallest of such indices, then $0 = x^\star_{k-1} < x^\star_k$ and $[x^\star_k, \cdots, x^\star_n]$ is the unique solution to $\Pi^{\text{iso}}_{\mathbb{R}}(\mathbf{y}_{k\to n})$ from Lemma 1. Then $[x^{\text{iso}}_1, \cdots, x^{\text{iso}}_{k-1}, x^\star_k, \cdots, x^\star_n]$ is also feasible for (17) ($x^{\text{iso}}_{k-1} \le l < x^\star_k$) but has lower objective value than $\mathbf{x}^{\text{iso}}$, this contradicts the unicity of the solution. Hence $x^\star_k = 0, \forall k$.

General case: We suppose $\exists k \in \{1, \cdots, n\}$ such that: $x^{\text{iso}}_{k-1} \le 0 < x^{\text{iso}}_k$. From Lemma 1, $[x^{\text{iso}}_1, \cdots, x^{\text{iso}}_{k-1}]$ and $[x^{\text{iso}}_k, \cdots, x^{\text{iso}}_n]$ are then solutions to $\Pi^{\text{iso}}_{\mathbb{R}}(\mathbf{y}_{1\to k-1})$ and $\Pi^{\text{iso}}_{\mathbb{R}}(\mathbf{y}_{k\to n})$ respectively. From above, the vector $[0, \cdots, 0] \in \mathbb{R}^{k-1}$ is solution to $\Pi^{\text{iso}}_{[0,+\infty)}(\mathbf{y}_{1\to k-1})$. Then the global vector $[0, \cdots, 0, x^{\text{iso}}_k, \cdots, x^{\text{iso}}_n]$ is the solution to the global program:

$$\min \sum_{i=1}^{n} (y_i - x_i)^2 \quad \text{s.t.} \quad 0 \le x_1 \le \cdots \le x_{k-1}, \quad x_k \le \cdots \le x_n$$

Adding the constraint $x_{k-1} \le x_k$ does not affect the solution. Hence $[0, \cdots, 0, x^{\text{iso}}_k, \cdots, x^{\text{iso}}_n]$ is the solution to $\Pi^{\text{iso}}_{[0,+\infty)}(\mathbf{y})$.□

B. Efficient Implementation of the PAVA

Since our code base is in Python, we use the PAVA from `scikit-learn`. However, as observed by Tulloch,[1] `scikit-learn`'s implementation is rather slow compared, *e.g.*, to the PAVA in R. Even though the original algorithm has $O(n)$ complexity where n is the dimension of the problem, PAVA from `scikit-learn` scales close to $O(n^2)$ due to the cost of maintaining active sets. Hence, we write our own C++ implementation of Tulloch's algorithm, called `PAVA+(·)`, and use Cython[2] to build the wrappers for Python. The algorithm detects a decreasing subsequence $y_i, \cdots, y_k$ and replaces each point in the subsequence by the average of the subsequence to minimize the distance to $\mathbf{y}$, the vector we project, while satisfying the order constraint. By implementing entirely in-place, Tulloch obtained an algorithm 5,000 times than `scikit-learn`. Our C++ implementation is 16,000 times faster than `scikit-learn`.[3]

The algorithm always iterates through all the entries of $\mathbf{y}$ at each execution of the main loop, but it would be more efficient to skip constant subsequences[4] formed in previous iterations by substituting decreasing subsequences by their average. When solving the ℓ_1-constrained problem (1), the iterates $\mathbf{x}_t$ can be sparse due to the ℓ_1 regularization, where sequences of zeroes $0 = x_i$ for $i = r, r+1, \cdots, s$ translates into constant sequences $z_{r-1} = z_r = \cdots = z_s$ (recall that $0 = z_i - z_{i-1}$ from (14)). Hence, we add an integer array $\mathbf{w} \in$

[1] http://tullo.ch/articles/speeding-up-isotonic-regression/

[2] Cython is a popular language for compiling Python into plain C by adding static type declarations, see http://cython.org/

[3] For 1,000,000 points from a perturbed $\log(1+x)$, we perform in 43.4ms while `scikit-learn` finishes in 690s on average.

[4] In the isotonic regression literature, such constant subsequences are commonly called active sets.

$\mathbb{N}^n$ initialized with all ones, and update the entries w_i by the size of the constant subsequence starting at index i. We still preserve the efficiency of PAVA+ $(\cdot)$ with only in-place operations on $\mathbf{y}$ and $\mathbf{w}$ but with only one replacement per constant subsequence. The pseudocode is presented below:

Algorithm 1 PAVA++ $(\cdot)$ Improved PAVA algorithm.

$\mathbf{w} := [1, \cdots, 1] \in \mathbb{R}^n$
pooled:$= 1$
while pooled$== 1$:
    pooled:$= 0$
    $i := 1$
    while $i \leq n$:
        $k := i + w_i$
        $j := i$
        while $k \leq n$ and $y_k \leq y_j$:
            $j := k$
            $k := k + w_k$
        if $y_i \neq y_j$:
            numerator:$= 0$
            denominator:$= 0$
            $j := i$
            while $j < k$:
                numerator :=numerator $+ w_j y_j$
                denominator := denominator $+ w_j$
                $j := j + w_j$
            y_i :=numerator / denominator
            w_i :=denominator
            pooled:$= 1$
        $i := k$

As shown in Figure 1, numerical experiments on randomly perturbed points $\log(i)$, $i = 1, \cdots, n$ show that PAVA++ $(\cdot)$ is 60% times faster than PAVA+ $(\cdot)$, from 5 to 10 times faster than Tulloch's PAVA, and 24,000 times faster then scikit-learn's PAVA. We also separate the points $\log(i)$, $i = 1, \cdots, n$ into K blocks $\{r_k, r_k + 1, \cdots, r_{k+1} - 1\}, k = 1, \cdots, K$, each point in the same block being set to z_{r_k}, the first point of the block. Hence we construct uniform subsequences which correspond to a sparse vector $\mathbf{x}$ with non-zero entries equal to $x_{r_k} = z_{r_k} - z_{r_k - 1}$, the difference of values between consecutive uniform subsequences, see equation (14). When computations are carried out in a sparse setting, we know the non-zero entries of $\mathbf{x}$ and hence the partition $\{r_k, r_k + 1, \cdots, r_{k+1} - 1\}, k = 1, \cdots, K$. PAVA++ $(\cdot)$ is 10 times faster as illustrated in Figure 1, as a result of initializing the weight vector with the partition block sizes in Algorithm 1.

V. Implementation and Experiments

A. Efficient implementation of subgradient methods

For our numerical results, we also wrote optimized C++/Cython implementations of the projection onto the simplex S^n_λ, the projection onto the ℓ_1-ball B^λ_+, and the projection onto the product of simplexes $S^{n_1}_{\lambda_1} \times \cdots \times S^{n_s}_{\lambda_s}$.[5] We compare the rates of converence of the projected gradient method [3] with the Barzilai-Borwein step size selection [2], and the L-BFGS (see [15]), augmented with a backtracking

[5]The code is open source and available at https://github.com/megacell/block-simplex-least-squares

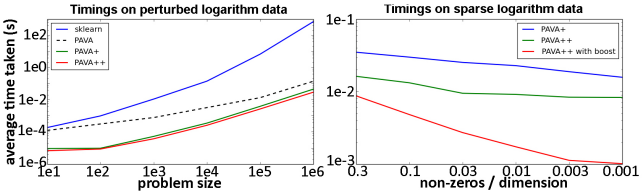

Fig. 1. **Left: comparison between the computation times averages over 10 trials of PAVA from scikit-learn (sklearn), Tulloch's PAVA, its C++ implementation (PAVA+), and with constant sequences tracking (PAVA++) on logarithm data. Right: comparison between the computation times of the PAVA+, PAVA++, and PAVA++ with prior knowledge on the constant subsequences.**

line search [5, §9.2]. Our implementation of these methods are mostly in Python, and the matrix manipulations relies heavily on the NumPy package, as well as SciPy.sparse for sparse matrices multiplications. Cython has been used to wrap the projections implemented in C++.

Our experiments consider the least squares setting with design matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, observations $\mathbf{b} \in \mathbb{R}^m$, and parameters $\mathbf{x} \in \mathbb{R}^n$, where we aim to minimize the least-squares objective $f(\mathbf{x}) = \frac{1}{2}\|A\mathbf{x} - \mathbf{b}\|_2^2$ subject to the simplex constraint $\mathbf{x} \in S^n_\lambda$, that is n is the number of parameters and m the number of observations.

B. Linear regression model

We have four setups in which the entries of $\mathbf{A}$ are i.i.d. samples from 1) the normal distribution, 2) the log-normal distribution, 3) the exponential distribution, and in the fourth setup we have the "cumulative normal":

$$A_{ij} = \sum_{k=j}^{n} a_{ik} \quad \text{with} \quad a_{ij} \overset{\text{i.i.d.}}{\sim} N(0,1), \forall i, j \tag{20}$$

The parameters $\mathbf{x}$ are drawn blockwise from Dirichlet distributions with hyperparameter α, and the simplex scaling factor λ is drawn uniformly on $[0, \Lambda]$. Then the measurements are generated via $\mathbf{b} = \mathbf{A}\mathbf{x}$. We set $n = 1000$ and compare the difference of performance between solving the simplex-constrained problem (2) and the equivalent re-formulation via our affine transformation $\mathbf{x} = \mathbf{N}\mathbf{z} + \mathbf{e}$. The error $f(\mathbf{x}) - f(\mathbf{x}^\star)$ typically presents a linear convergence, as illustrated in Figure 2. We get a proxy for the log of the convergence rate by doing a linear fit of the accuracy.

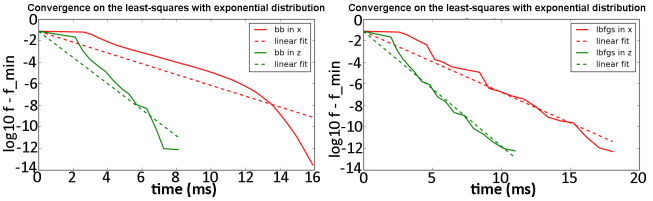

Fig. 2. **Accuracies $f(\mathbf{x}) - f(\mathbf{x}^\star)$ as a function of time in log scale for A_{ij} sampled from the exponential distribution. We have approximately linear convergence with respect to time.**

The empirical estimates of the log rates are aggregated in Figure 3.a. as a function of the number of measurements $m = 10, 30, 100, 300, 1000$. For $m = 1000 (= n)$, the matrix

	normal	cumulative	exponential	log-normal
$c(\mathbf{A}^T\mathbf{A})$	611	5.71×10^8	7.56×10^5	2.72×10^6
$c((\mathbf{AN})^T\mathbf{AN})$	1.48×10^8	2.42×10^3	1.29×10^8	5.35×10^7
coherence $\mathbf{A}$	0.153	0.99	0.587	0.606
coherence $\mathbf{AN}$	0.188	0.14	0.196	0.506
mean corr. $\mathbf{A}$	0.0252	0.558	0.502	0.374
mean corr. $\mathbf{AN}$	0.0309	0.0252	0.0309	0.0302

TABLE I
CONDITION NUMBER OF $\mathbf{A}^T\mathbf{A}$ AND $(\mathbf{AN})^T\mathbf{AN}$ FOR $m = 1000$, AND AVERAGE MUTUAL COHERENCE, AND CROSS-CORRELATION OF $\mathbf{A}$ AND $\mathbf{AN}$ FOR $m = 10, 30, \cdots, 1000$ FOR THE REGRESSION MODEL.

	aggregated	uniform	highway network
$c(\mathbf{A}^T\mathbf{A})$	4.60×10^7	9.00×10^5	
$c((\mathbf{AN})^T\mathbf{AN})$	7.95×10	4.84×10^3	
coherence $\mathbf{A}$	0.900	0.418	0.675
coherence $\mathbf{AN}$	0.222	0.191	0.476
mean corr. $\mathbf{A}$	0.659	0.300	0.275
mean corr. $\mathbf{AN}$	0.0315	0.0314	0.181

TABLE II
CONDITION NUMBER OF $\mathbf{A}^T\mathbf{A}$ AND $(\mathbf{AN})^T\mathbf{AN}$ FOR $m = 1000$, AND AVERAGE MUTUAL COHERENCE, AND CROSS-CORRELATION OF $\mathbf{A}$ AND $\mathbf{AN}$ FOR $m = 10, 30, \cdots, 1000$ FOR THE SIGNAL RECOVERY PROBLEM.

$\mathbf{A}$ is full rank, so we can compute the condition number of the Hessian $\nabla^2 f = \mathbf{A}^T\mathbf{A}$ of f, that is

$$c(\mathbf{A}^T\mathbf{A}) = \lambda_{\max}(\mathbf{A}^T\mathbf{A})/\lambda_{\min}(\mathbf{A}^T\mathbf{A}) \tag{21}$$

which directly affects the rate of convergence since the convergence is at least linear for (unconstrained) minimization of strongly convex functions [5, §9.3.1]:[6]

$$\frac{f(\mathbf{x}_t)-f(\mathbf{x}^\star)}{f(\mathbf{x}_0)-f(\mathbf{x}^\star)} \leq \kappa(\mathbf{A}^T\mathbf{A})^t = \left(1-\frac{1.6\times10^{-4}}{c(\mathbf{A}^T\mathbf{A})}\right)^t \tag{22}$$

Hence we have faster convergence when the condition number $c(\mathbf{A}^T\mathbf{A})$ is close to 1. The condition number after affine transformation $\mathbf{x} := \mathbf{Nz}+\mathbf{e}$ on the least-squares is $c((\mathbf{AN})^T\mathbf{AN})$, hence we compare both values in Table 1. When $m = 1000$, the condition number is lower in the $\mathbf{x}$-basis for all the distributions except the cumulative normal. Hence there is faster convergence for these distributions, as illustrated in Figure 3.a. and predicted by (22). For the cumulative normal, the product $\mathbf{AN}$ has exactly entries $A_{ij}-A_{i,j+1}$, hence we recover exactly the samples a_{ij} in (20). Learning directly on the points a_{ij} enables a faster convergence with the $\mathbf{z}$ variables as seen in Figure 3.a.

On the contrary, the gradient descent methods converge faster with the $\mathbf{z}$ variables for sparse data ($m \leq 100$) generated from skewed distributions (exponential and log-normal here). Let us define the average cross-correlation and the mutual coherence of $\mathbf{A}$, which where introduced by Donoho and Huo [7] and has been used extensively in the field of sparse representations of signals:

$$\begin{aligned}&\text{average cross-correlation of } \mathbf{A} = \text{mean}\{|\mathbf{a}_i^T\mathbf{a}_j|\}_{i\neq j}\\ &\text{mutual coherence of } \mathbf{A} = \max\{|\mathbf{a}_i^T\mathbf{a}_j|\}_{i\neq j}\end{aligned} \tag{23}$$

where $\mathbf{a}_i$ denotes the normalized rows of $\mathbf{A}$. Table 1 shows that the average cross-correlation and the mutual coherence are smaller for $\mathbf{AN}$ when data points are sampled from skewed distribution, suggesting that the $\mathbf{z}$ variables form a better basis, the extreme case being the cumulative normal in which we fit on the samples $a_{ij} \sim N(0,1)$ directly.

[6] In the general case, we have $c = 1-2\lambda_{\min}\alpha\min(1,\beta/\lambda_{\max})$, where α and β are the tuning parameters of the backtracking line search. We choose $\alpha = 10^{-4}$ and $\beta = 0.8$ which reduces c to the expression above.

C. Sparse signal recovery from linear measurements

For these numerical experiments, we focus on block-simplex least-squares problems (2) of the form

$$\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{Ax}-\mathbf{b}\|_2^2 \quad \text{s.t.} \quad \mathbf{x} \in S^{n_1}_{\lambda_1}\times\cdots\times S^{n_s}_{\lambda_s} \tag{24}$$

and we want to recover the signal $\mathbf{x}$ from sparse linear measurements $\mathbf{a}_i^T\mathbf{x}$, where $\mathbf{a}_i^T \in \{0,1\}^n$ for all $i = 1,\cdots,m$. Hence $\mathbf{A} \in \{0,1\}^{m\times n}$ is an incidence matrix that maps the state $\mathbf{x}$ (to be estimated) of a physical system to aggregate measurements of it. This setting appears widely in physical systems which can be modeled as a graph, such as in transportation networks [19].

We consider a first model in which all the entries of $\mathbf{A}$ are i.i.d samples from a Bernoulli distribution with constant parameter p (we take $p = 0.3$ to emulate sparsity) and a second model that emulates measurements which are correlated with the block structure of the parameter vector. In the second model, the generation of A_{ij} is no longer i.i.d. Now instead, the rows $\mathbf{a}_i^T \in \mathbb{R}^n$ are generated independently based on a pivot k randomly selected in $\{1,\cdots,n\}$ which determines the Bernoulli success probabilities for the entire row: $p_i = 0.9$ for $|i-k| \leq K$, and 0.1 otherwise, resulting in clusters of ones in the overall design matrix, all other entries being set to 0.

We apply the same projected subgradient methods and compare their rates of convergence on the two models described above. When the measurements are sparse ($m \leq 100$), the algorithms are faster in the $\mathbf{z}$-basis as shown in Figure 3.b. Table 2 shows that the average cross-correlation and the mutual coherence are smaller for $\mathbf{AN}$ for both models, suggesting that the observation model $\mathbf{AN}$ generates sparse measurements that are less correlated, hence the superior convergence for our projected gradient descent methods.

D. Application to route flow estimation

For our application, we consider the highway network near Los Angeles composed of 44 nodes and 122 directed arcs, see Figure 4. The roads' characteristics are obtained from OpenStreetMaps, the (Origin-Demand) OD demands are based on data from the Census Bureau. They represent a quasi-static morning rush hour model. The experimental setup is identical to [19]. We compute the equilibrium flow using the (Bureau of Public Roads) BPR-type delay function

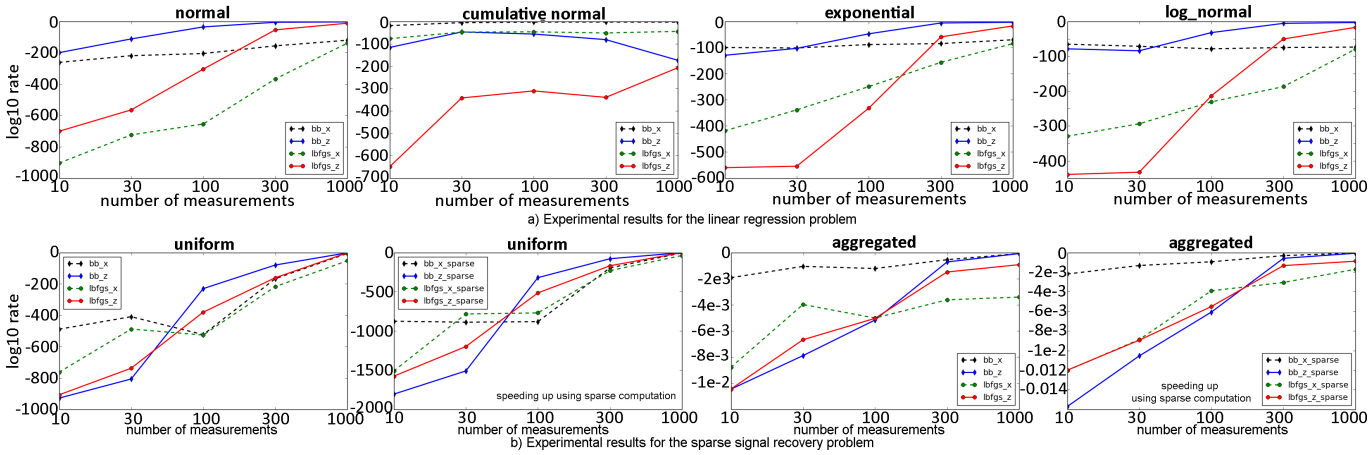

Fig. 3. **Comparison of the methods on a) the linear regression model and b) the sparse signal recovery problem. The dimension $n = 1000$ and the number of measurements varies from 10 to 1000 (full rank).**

$s_a^{\text{BPR}}(v_a) = d_a(1+0.15(v_a/m_a)^4)$ with v_a, d_a and m_a the flow, free flow delay and capacity on arc a and then enumerate the used routes P between OD pairs using a k-shortest paths algorithm. We suppose we have measurements from loop detectors mainly along the I-210, I-10 and state route 60, see Figure 4. We are interested in estimating the route flows x_p for $p \in P$, and pose it as a least-squares problem of the form (24). The projected gradient methods are faster with the **z** variables and Table 2 shows that **AN** has lower mutual coherence and average cross-correlation.

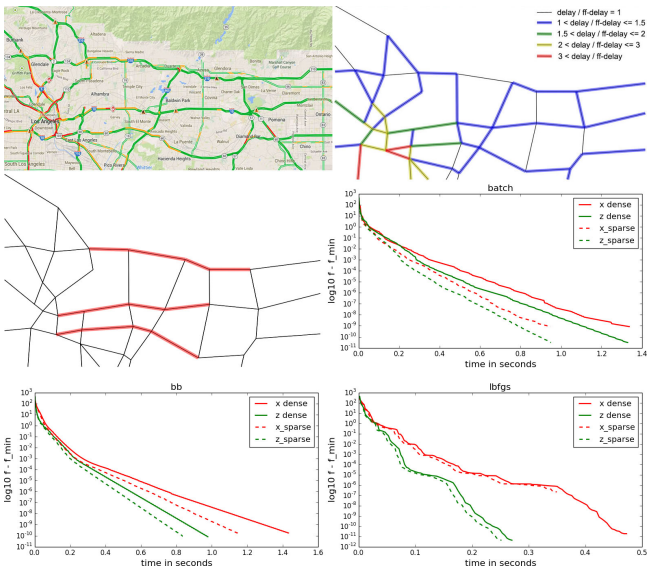

Fig. 4. **Application to the highway network of Los Angeles. Top left: typical morning rush hour on 2014- 06-12 at 9:14 AM from Google Maps. Top right: delays from the User Equilibrium. Middle left: observed links.**

REFERENCES

[1] R. E. Barlow, D. Bartholomew, J. M. Bremner, and H. D. Brunk. *Statistical inference under order restrictions; the theory and application of isotonic regression*. Wiley, New York, 1972.

[2] Jonathan Barzilai and Jonathan M. Borwein. Two-Point Step Size Gradient Methods. *IMA Journal of Numerical Analysis*, 8:141–148, 1988.

[3] D. P. Bertsekas. *Network Optimization: Continuous and Discrete Methods*. Athena Scientific, Belmont, MA 02178, 1998.

[4] Michael J. Best and Nilotpal Chakravarti. Active set algorithms for isotonic regression; a unifying framework. *Mathematical Programming*, 47:425–439, 1990.

[5] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, March 8 2004.

[6] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to algorithms*. MIT Press, Cambridge, MA, USA, 2001.

[7] D. L. Donoho and X. Huo. Uncertainty principles and ideal atomic decomposition. *IEEE Transactions on Information Theory*, 47:28452862, 2001.

[8] J. Duchi, S. Gould, and D. Koller. Projected subgradient methods for learning sparse gaussians. *Proceedings of the 24th Conference on Uncertainty in Artificial Intelligence*, 2008.

[9] John Duchi, Shai Shalev-Shwartz, Yoram Singer, and Tushar Chandra. Efficient Projections onto the l1-Ball for Learning in High Dimensions. *Proceedings of the 25th International Conference on Machine Learning*, 2008.

[10] Thore Graepel, Joaquin Quinonero Candela, Thomas Borchert, and Ralf Herbrich. Web-Scale Bayesian Click-Through Rate Prediction for Sponsored Search Advertising in Microsofts Bing Search Engine. *ICML*, 2009.

[11] S. J. Grotzinger and C. Witzgall. Projection onto Order Simplexes. *Applied Mathematics and Optimization*, 12:247–270, 1984.

[12] D. N. Joanes and C. A. Gill. Comparing measures of sample skewness and kurtosis. *Journal of the Royal Statistical Society*, 47:183189, 1998.

[13] Anastasios Kyrillidis, Stephen Becker, Volkan Cevher, and Christoph Koch. Sparse projections onto the simplex. *arXiv:1206.1529*, April 2013.

[14] H. Brendan McMahan, Gary Holt, D. Sculley, Michael Young, Dietmar Ebner, and Julian Grady. Ad Click Prediction: a View from the Trenches. *KDD*, 2013.

[15] J. Nocedal and S. Wright. *Numerical Optmization*. Springer, 2nd edition, 2006.

[16] M. Patriksson. *The Traffic Assignment Problem - Models and Methods*. VSP, Utrecht, 1994.

[17] Mert Pilanci, Laurent El Ghaoui, and Venkat Chandrasekaran. Recovery of Sparse Probability Measures via Convex Programming. *Neural Information Systems foundation (NIPS)*, 2012.

[18] R. J. Tibshirani, H. Hoefling, and R. Tibshirani. Nearly-Isotonic Regression. *Technometrics*, 53, 2011.

[19] C. Wu, J. Thai, S. Yadlowsky, A. Pozdnukhov, and A. Bayen. Cellpath: fusion of cellular and traffic sensor data for route flow estimation via convex optimization. *21st International Symposium on Transportation and Traffic Theory (ISTTT)*, 2015.