

Fundamentals of cone regression

Mariella Dimiccoli

Edifici O, Campus UAB
08193, Bellaterra (Cerdanyola del Vallès), Spain
e-mail: dimiccolimariella@gmail.com

Abstract: Cone regression is a particular case of quadratic programming that minimizes a weighted sum of squared residuals under a set of linear inequality constraints. Several important statistical problems such as isotonic, concave regression or ANOVA under partial orderings, just to name a few, can be considered as particular instances of the cone regression problem. Given its relevance in Statistics, this paper aims to address the fundamentals of cone regression from a theoretical and practical point of view. Several formulations of the cone regression problem are considered and, focusing on the particular case of concave regression as an example, several algorithms are analyzed and compared both qualitatively and quantitatively through numerical simulations. Several improvements to enhance numerical stability and bound the computational cost are proposed. For each analyzed algorithm, the pseudo-code and its corresponding code in Matlab are provided. The results from this study demonstrate that the choice of the optimization approach strongly impacts the numerical performances. It is also shown that methods are not currently available to solve efficiently cone regression problems with large dimension (more than many thousands of points). We suggest further research to fill this gap by exploiting and adapting classical multi-scale strategy to compute an approximate solution.

MSC 2010 subject classifications: Primary 62; secondary 90.

Keywords and phrases: cone regression, linear complementarity problems, proximal operators.

Received March 2015.

Contents

1	Introduction	54
2	Statement of the problem, basic notations and basic facts	55
2.1	Convex quadratic programming (CQP) formulation	57
2.1.1	Primal formulation	57
2.1.2	Dual formulation	57
2.2	Linear complementarity problem (LCP) formulation	59
2.3	Proximal formulation	59
3	State of the art	60
3.1	Algorithms with asymptotic convergence	61
3.1.1	Least squares in a product space (LSPS)	61
3.1.2	Uzawa method	62
3.1.3	Hildreth’s algorithm	62
3.1.4	Primal-dual interior point methods	62
3.1.5	Dykstra’s algorithm	63

3.1.6	Alternating Direction Method of Multipliers (ADMM) . . .	64
3.2	Algorithms with finite-time convergence	65
3.2.1	Properties of polyhedral convex cones with $m \leq n$	65
3.2.2	Early algorithms based on the properties of polyhedral convex cones	67
3.2.3	Mixed primal-dual basis algorithm	68
3.2.4	Critical index algorithm: Nearest point problem in sim- plicial cones	69
3.2.5	Meyer's algorithm	72
4	Issues about effectiveness for large-scale problems	73
4.1	Suitability to take advantage of available good estimates	73
4.1.1	PAV's inspired approximate solution	75
4.2	Suitability to be implemented in an online fashion	75
4.3	Computational issues	75
5	Improving the active set framework for concave regression problems	77
6	Comparative results	78
7	Conclusions	80
A	Appendix A	80
A.1	Signals used for comparative evaluations	80
A.2	Comparative evaluations on signals of size 50	82
A.3	Comparative evaluations on signals of size 500	85
A.4	Comparative evaluations on signals of size 1000	88
B	Pseudocode of the algorithms reviewed	91
	Acknowledgements	96
	Supplementary Material	96
	References	96

1. Introduction

Cone regression analysis is a valuable alternative to more traditional parametric-regression models, in all cases where the functional relationships between the response (dependent) and the explanatory (independent) variables is unknown and nonlinear and the constraints are a set of linear inequalities. Several important statistical problems including isotonic, concave and constrained spline regression, or ANOVA under partial orderings can be seen as particular instances of the more general cone regression problem. Cone regression admits several formulation approaches and implementation strategies, whose choice severely impacts numerical performances. However, due to the little exposure to the topics of optimization theory in modern-day Statistics, many optimization and numerical approaches are commonly ignored by statisticians. This paper is a contribution to fill this gap in the literature, addressing the fundamentals of cone regression from a theoretical and practical point of view. With the goal of going in deep with comparisons and numerical issues, we focus in particular on the concave regression problem. In spite of its theoretical simplicity, since the number of constraints increases linearly with the data size, concave regression

offers a good basis to discuss the fundamentals of cone regression and related numerical issues.

The problem of concave regression is to estimate a regression function subject to concavity constraints represented by a set of linear inequalities. Brought to the attention of the scientific community by micro-economists interested in estimating production functions [1, 2, 3], the problem of concave regression arises not only in the field of micro-economy (indirect utility, production or cost functions, Laffer curve) but also in medicine (dose response experiments) and biology (growth curves, hazard and failure rate in survival analysis). First addressed by Hildreth in 1954 [1], the search for efficient methods for solving large concave regression problems is still an open issue. This may appear quite surprising considering the noticeable advances in convex optimization since then, but it can be understood when considering that most efforts have been devoted to theoretical issues such as generalizations and convergence while comparatively little attention has been paid to the issues of efficiency and numerical performances in practice [4, 5, 6].

In this paper, we formulate the cone regression problem by different optimization approaches. We highlight similarities and difference between the various algorithms passed in review, we propose several improvements to enhance stability and to bound the computational cost. We estimate the expected performance of available algorithms, establishing in particular which is the most competitive technique for solving large instances of the problem. Finally, in the light of this study, we give recommendations for further research.

In section 2, we state formally the problem of cone regression also introducing some basic notations and results that will be used throughout the paper. In section 3, we survey the state of the art distinguishing between the class of algorithms with asymptotic convergence and the class of algorithms with finite-time convergence. In section 6, we make a numerical comparison of performances and finally, in section 7, we draw some concluding remarks.

2. Statement of the problem, basic notations and basic facts

The aim of a regression analysis is to produce a reasonable analysis to the unknown response function f , which can be modeled as

$$y = f(z) + \epsilon \quad (2.1)$$

where $z \in \mathcal{R}$ is the explanatory (dependent) variable, $y \in \mathcal{R}^d$ is the response (independent) random variable, and ϵ is an error term, which is usually assumed to be a mean zero random variable. Typically, one has observations on y and z for n selected values of z . For each level of input, say z_i , there may be several trials and corresponding observations of output y_i . Let T_i be the number of trials at level of input z_i and let y_{it} be the observed output for the t -trial at this level. Then we have

$$y_{it} = f(z_i) + \epsilon_{it}, \quad i = 1, \dots, n \quad t = 1, \dots, T_i \quad (2.2)$$

Inference about the response function may be drawn by assuming that the function $f(z)$ can be approximated by some given algebraic form with several unknown parameters to be estimated from the data. However, the difficulty with this procedure is that the inferences often depend critically upon the algebraic form chosen. Alternatively, one may know some properties of the relation being studied but does not have sufficient information to put the relation into any simple parametric form. In this case, a nonparametric approach is more appropriate. Let x_i be the expected value of output at input level z_i :

$$x_i = f(z_i) \quad i = 1, \dots, n \quad (2.3)$$

Estimates of x_i can be derived by the method of maximum likelihood, or by the method of least squares or other formulations. If there were no a priori restriction on f , than the maximum likelihood estimation of x_i would just be the mean of observed output for the level of input z_i , that is

$$\tilde{x}_i = \frac{1}{T_i} \sum_{T_i}^{i=1} y_{it} \quad i = 1, \dots, n \quad (2.4)$$

Instead, since in the cone regression problem the known property of the regression function f can be expressed by a set of linear inequalities, to obtain the maximum likelihood estimates, the likelihood function should be maximized subject to the linear inequality constraints.

Formally, given a dataset of n dependent variables represented by the vectors $w, y \in \mathcal{R}^n$, corresponding to the independent variable values $z_1 < z_2 < \dots < z_n$, the problem of cone regression is to estimate the closest function to the dataset via a least squares regression subject to a set of linear inequality constraints by solving

$$\hat{x} = \underset{\{x'' \leq 0\}}{\operatorname{argmin}} \|x - y\|_{2,w}^2 \quad (2.5)$$

$$\text{with } \|x - y\|_{2,w}^2 = \sum_{i=1}^n w_i (y_i - x_i)^2$$

Denoting by \mathcal{K}_i those vectors that satisfy the linear inequality constraints for a fixed i , then $\mathcal{K}_i \neq \emptyset$ is a closed convex set in \mathcal{R}^n and the feasibility set \mathcal{K} can be written as the nonempty intersection of a family of closed subsets $\mathcal{K}_i \subset \mathcal{R}^n$. Being the intersection of closed convex sets, the set \mathcal{K} is also a closed convex set. More precisely, since each \mathcal{K}_i is an half-space which contains the origin, the feasibility set \mathcal{K} is a convex polyhedral cone. In matrix form, \mathcal{K} can be written as $\mathcal{K} = \{x : Ax \leq 0\}$. In the case of concave regression $A \in \mathcal{R}^{m \times n}$ with $m = n - 2$ is a matrix such that each row A_i represents a concave two-piece linear function with a negative second difference at x_{i+1} only and the linear inequalities are as follows

$$\frac{x_{i+2} - x_{i+1}}{z_{i+2} - z_{i+1}} - \frac{x_{i+1} - x_i}{z_{i+1} - z_i} \leq 0, \quad i = 1, \dots, n - 2 \quad (2.6)$$

In the following, we give alternative formulations of the cone regression problem that rest on optimization theory.

2.1. Convex quadratic programming (CQP) formulation

2.1.1. Primal formulation

The problem (2.5) is to find the point \hat{x} in the cone \mathcal{K} that is closest to y . The solution is found at the orthogonal projection of y onto \mathcal{K} , written as $\Pi(y|\mathcal{K})$ using the metric $\|\cdot\|_{2,w}$, represented by the symmetric positive definite matrix W .

$$\hat{x} = \Pi(y|\mathcal{K}) = \underset{\{Ax \leq 0\}}{\operatorname{argmin}} (y-x)^T W (y-x) \quad (2.7)$$

For the problem (2.7), the matrix W is diagonal with element w_i on the diagonal. In practice, if y_i is the mean value measured at z_i , than w_i corresponds to the size of the sample at z_i . Since \mathcal{K} is a closed, convex and non empty set on the Hilbert space \mathcal{R}^n , it is a *set of Chebyshev*, that is the projection exists and it is unique.

2.1.2. Dual formulation

The dual formulation of problem (2.7) rests on the Moreau decomposition theorem [7], which is a generalization in convex analysis of the orthogonal projection theorem for vectorial sub-spaces. Central to the Moreau decomposition theorem is the definition of *polar cone* of a given convex cone \mathcal{K} , which is given below.

Definition 1. The *polar cone* \mathcal{K}° to any convex cone \mathcal{K} is given by

$$\mathcal{K}^\circ = \{x \in \mathcal{R}^n : \forall k \in \mathcal{K}, \langle k', x \rangle \leq 0\} \quad (2.8)$$

The Moreau decomposition theorem is as follows.

Theorem 2.1. Let $\mathcal{K} \subseteq \mathcal{R}^n$ be a closed convex cone, \mathcal{K}° its polar cone and $y \in \mathcal{R}^n$ a given point. Then the following assertions are equivalent:

$$\begin{aligned} (i) \quad & \hat{x} = \underset{x \in \mathcal{K}}{\operatorname{argmin}} \|x - y\|^2, \quad \hat{x}^\circ = \underset{x \in \mathcal{K}^\circ}{\operatorname{argmin}} \|x - y\|^2 \\ (ii) \quad & \hat{x} \in \mathcal{K}, \quad \hat{x}^\circ \in \mathcal{K}^\circ, \quad \langle \hat{x}, \hat{x}^\circ \rangle = 0, \quad y = \hat{x} + \hat{x}^\circ \end{aligned}$$

By relying on this theorem we can alternatively solve problem (2.7) by first finding the projection on the polar cone \hat{x}° and then computing the solution to the primal problem as the difference $\hat{x} = y - \hat{x}^\circ$ (see Figure 1). This alternative is attractive since, as it will be clarified below, it implies an analytically simpler form for the constraints.

Before stating an important Lemma about the relationship between the polar cone and the constraint matrix A , let us introduce the definition of *edges* of a polyhedral convex cone.

Definition 2. Let \mathcal{K} be a polyhedral convex cone in \mathcal{R}^n , then the vectors $e_i \in \mathcal{R}^n \setminus \{0\}$ are the *edges or generators* of \mathcal{K} if and only if $\mathcal{K} = \operatorname{pos}(\{e_i\}) = \{\sum k_i e_i | k \geq 0\}$.

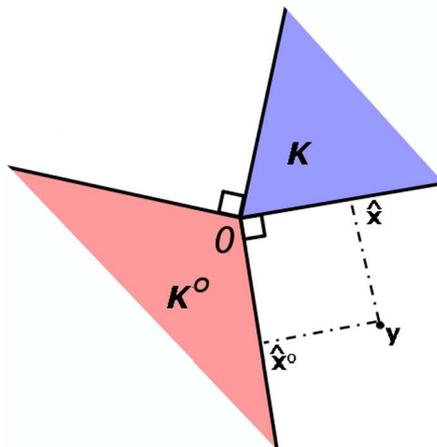


FIG 1. The polar cone \mathcal{K}^o of a given convex cone $\mathcal{K} \subset \mathcal{R}^2$ is given by the set of all vector whose scalar product with vectors of \mathcal{K} is negative. The data point y can be written as the sum of \hat{x} , the projection onto the cone \mathcal{K} and \hat{x}^o , the projection onto the polar cone \mathcal{K}^o .

Intuitively speaking, the edges of a polyhedral convex cone are one-dimensional rays, which always pass through a fixed point (the vertex).

Lemma 1. *The rows of the matrix A are the edges of the polar cone, that is $\mathcal{K}^o = \{x : x = \sum_{i=1}^m A_i^T a_i, a_i \geq 0\}$.*

To see that, observe that $\mathcal{K}^o = \{\sum_{i=1}^m a_i A_i^T, a_i \geq 0\}$ is polar to \mathcal{K} since $\forall x \in \mathcal{K}, \forall \rho \in \mathcal{K}^o: \langle \rho, x \rangle = \{\sum_{i=1}^m a_i \langle A_i^T, x \rangle \leq 0\}$, which is the definition of a polar cone of \mathcal{K} . Conversely, \mathcal{K} is polar to \mathcal{K}^o since: $\mathcal{K} = (\mathcal{K}^o)^o = \mathcal{K}$.

By relying on this result Khun-Tucker [8] proved the following theorem:

Theorem 2.2. *The primal constrained quadratic minimization problem (2.7) is equivalent to the dual problem*

$$\hat{\lambda} = \underset{\lambda \geq 0}{\operatorname{argmin}} (y - A^T \lambda)^T W (y - A^T \lambda) \quad (2.9)$$

Denoting by $\hat{\lambda}$ the solution to the dual problem, the solution to the primal problem is $\hat{x} = y - A^T \hat{\lambda}$.

As it can be observed, in the dual formulation each element of the vector λ must satisfy a single positivity constraint.

Goldman [9] noticed that dual problem can be also viewed as a minimum distance problem in the same parameter space as the primal problem.

$$\hat{x} = \underset{x \in \mathcal{C}}{\operatorname{argmin}} \|x\|^2 \quad (2.10)$$

where $\mathcal{C} = \{x | x = y - A^T \lambda, \lambda \geq 0\}$ is a rotation of the dual cone with its vertex translated to y . \hat{x} also solves the re-parametrized dual problem.

2.2. Linear complementarity problem (LCP) formulation

The CQP (2.7) can be recast as a linear complementarity problem (LCP). To see that, let us consider the Lagrangian associated with problem (2.7).

$$L(x, \lambda) = \|x - y\|_{2,w}^2 + \langle \lambda, Ax \rangle \quad (2.11)$$

where $\lambda \geq 0$ is the vector of dual variables associated to each of the convexity constraints. By applying the Karush–Kuhn–Tucker (KKT) optimality conditions [8] to (2.11), that is

$$\nabla L(\hat{x}, \hat{\lambda}) = 0 \quad (2.12)$$

$$\lambda \geq 0 \quad (2.13)$$

$$\lambda^T A\hat{x} = 0 \quad (2.14)$$

we obtain the equivalent LCP

$$w + M\lambda = q \quad (2.15)$$

$$w \geq 0, \quad \lambda \geq 0, \quad w^T \lambda = 0. \quad (2.16)$$

where $w = -A^T x$, $M = -AA^T$ and $q = -A^T y$. Note that by dropping the constant term from the Lagrangian and dividing it by 2: $L(x, \lambda) = \frac{1}{2}xx^T - y^T x + \lambda^T Ax$. Therefore: $\nabla L(x, \lambda) = x^T - y^T + \lambda^T A = 0$. By taking the transpose and multiplying by $-A$: $-Ax + (-AA^T)\lambda = -A^T y$. This LCP has a unique complementary solution. Denoting by $(\hat{w}, \hat{\lambda})$ its solution, $\hat{\lambda}$ is the optimal solution of the dual problem (2.9).

The condition $w^T \lambda = 0$ is called the *complementarity condition* and the way in which it is dealt with determines if the optimization algorithm belongs to the class of *interior point methods* that will be introduced in section 3.1.4 or to the class of *active set methods* that will be detailed in section 3.2.

2.3. Proximal formulation

The CQP (2.7) can be solved by using a proximity operator [10, 11]. Proximity operators are used to solve problems of the form

$$\operatorname{argmin}_{x \in \mathcal{R}^n} f_1(x) + f_2(x) \dots + f_m(x) \quad (2.17)$$

where f_1, f_2, \dots, f_m are convex functions from \mathcal{R}^n to $] -\infty, +\infty]$, that are not necessarily differentiable. Each f_i is treated through its proximity operator, which is defined as follows.

Definition 3. Let $\Gamma_0(\mathcal{R}^n)$ be the class of lower semicontinuous convex functions from \mathcal{R}^n to $] -\infty, +\infty]$ such that their domain, denoted by $\operatorname{dom}(f)$ is not the empty set. Let f be a function $f \in \Gamma_0(\mathcal{R}^n)$, then the *proximity operator* of f is $\operatorname{prox}_f(x) : \mathcal{R}^n \rightarrow \mathcal{R}^n$ such that

$$\forall x \in \mathcal{R}^n, \quad \operatorname{prox}_f(y) = \operatorname{argmin}_{x \in \mathcal{R}^n} f(x) + \frac{1}{2}\|x - y\|^2 \quad (2.18)$$

The proximity operator is characterized by the property

$$\forall(x, p) \in \mathcal{R}^n \times \mathcal{R}^n, \quad p = \text{prox}_f(y) \iff y - p \in \partial f(p), \quad (2.19)$$

where $\partial f : \mathcal{R}^n \rightarrow 2^{\mathcal{R}^n}$ is the subdifferential of f :

$$\partial f = \left\{ u \in \mathcal{R}^n : \forall y \in \mathcal{R}^n : (y - p)^T u + f(p) \leq f(y) \right\} \quad (2.20)$$

The proximity operator of a convex function is a generalization of the projection operator onto a closed convex set \mathcal{C} . To see that let us consider the indicator function of \mathcal{C}

$$\iota_{\mathcal{C}}(x) = \begin{cases} 0 & \text{if } x \in \mathcal{C} \\ +\infty & \text{if } x \notin \mathcal{C} \end{cases}$$

By using the fact that minimizing $J(x)$ over \mathcal{C} is equivalent to minimizing $J(x) + \iota_{\mathcal{C}}(x)$ over \mathcal{R}^n

$$\underset{x \in \mathcal{C}}{\operatorname{argmin}} J(x) = \underset{x \in \mathcal{R}^n}{\operatorname{argmin}} \{J(x) + \iota_{\mathcal{C}}(x)\}$$

it results that $\operatorname{prox}_{\iota_{\mathcal{C}}} = \Pi(\cdot | \mathcal{C})$.

The solution to problem (2.7) can be therefore understood as the proximity operator over \mathcal{K}

$$\hat{x} = \underset{x \in \mathcal{K}}{\operatorname{argmin}} \|x - y\|^2 = \underset{x \in \mathcal{R}^n}{\operatorname{argmin}} \{\|x - y\|^2 + \iota_{\mathcal{K}}(x)\} \quad (2.21)$$

$$= \operatorname{prox}_{\iota_{\mathcal{K}}} y \quad (2.22)$$

Alternatively, using the fact that $(\mathcal{K}_1 \cap \dots \cap \mathcal{K}_m)^o = \mathcal{K}_1^o + \dots + \mathcal{K}_m^o$, the dual problem (2.7) can be seen as the proximity operator over the sum of m indicator functions of the convex sets \mathcal{K}_i^o :

$$\hat{x}^o = \underset{x \in \sum_{i=1}^m \mathcal{K}_i^o}{\operatorname{argmin}} \|x - y\|^2 = \underset{x \in \mathcal{R}^n}{\operatorname{argmin}} \{\|x - y\|^2 + \sum_{i=1}^m \iota_{\mathcal{K}_i^o}(x)\} \quad (2.23)$$

$$= \operatorname{prox}_{\sum_{i=1}^m \iota_{\mathcal{K}_i^o}} y \quad (2.24)$$

Intuitively, the base operation of a proximal algorithm is evaluating the proximal operator of a function, which itself involves solving a small convex optimization problem. These subproblems often admit closed form solutions or can be solved very quickly with standard or simple specialized methods.

3. State of the art

In this section, we review existing algorithms for solving the cone regression problem. The algorithmic approaches, and in turn their numerical performances, strongly depend on the choice of the problem formulation. All existing methods are iterative and they can attain the optimal solution since \mathcal{K} is a Chebyshev set and therefore the optimal solution must exist in this closed set. However,

in terms of their numerical performances they can be classified into two broad classes: the class of methods that never (or in very simple cases) attain the optimal solution [1, 12] and those of methods that converge to the optimal solution in a finite number of steps [13, 14, 15, 16, 17, 18, 19, 20]. As it will be clarified in the following, methods with asymptotic convergence rest on the properties of the sub-gradient or more in general of proximity operators and act by finding the solution as the limit of a sequence of successive approximations. They are typically derived from the primal, the dual or from the proximal formulation. Methods with finite-time convergence exploit the geometric properties of polyhedral convex cones and find the exact solution as a non-negative linear combination of functions, forming a basis in a specified finite dimensional space. They are typically derived from the linear complementarity problem formulation.

3.1. Algorithms with asymptotic convergence

This section includes algorithms based on the primal formulation such as Least Squares in a Product Space (section 3.1.1), algorithms based on the dual formulation such as the Uzawa's method (section 3.1.2) and Hildreth's method (section 3.1.3), algorithms that solve the dual problem simultaneously with the primal problem such as the Dykstra's alternating projection method (section 3.1.5) and algorithms based on the proximal formulation such as Alternating Direction Method of Multipliers (section 3.1.6).

3.1.1. Least squares in a product space (LSPS)

Since the Euclidean space \mathcal{R}^n equipped with the dot product is an Hilbert space \mathcal{H} , the problem (2.7) can be recast in the m -fold Cartesian product of \mathcal{H} , say \mathcal{H}^m [35]. Let \mathcal{K}^m be the Cartesian product of the sets $(K_i)_{i \in I}$, i.e., the closed convex set $\mathcal{K}^m = \times_{i \in I} \mathcal{K}_i = \{x \in \mathcal{H} : \forall i \in I : x_i \in \mathcal{K}_i\}$ and let D be the diagonal vector subspace, i.e. $D = \{(x, \dots, x) \in \mathcal{H}^m : x \in \mathcal{H}\}$. Then, the CQP (2.7) is equivalent to

$$\operatorname{argmin}_{\{x \in \mathcal{K}^m \cap D\}} \|x - \bar{y}\|_{2,w}^2 \quad (3.1)$$

where $\bar{y} = (y, \dots, y)$. Using this strategy, the problem of projecting onto the intersection of m convex sets is reduced to the problem of projecting, in a higher dimensional space, onto only two convex sets, one of which is a simple vector subspace. Geometrically, this is equivalent to find a point in D which is at minimum distance from \mathcal{K}^m . This point can be obtained iteratively by

$$x_{k+1} = x_k + \lambda_k (P_D \circ P_{\mathcal{K}^m}^m(x_k) - x_k) \quad (3.2)$$

The advantage of this strategy is that it speeds up the convergence since a bigger (than 2, which is the upper bound for Fejér sequences [36]) relaxation interval can be allowed.

3.1.2. Uzawa method

A classical method to solve a convex minimization problem subject to inequality constraints is the Uzawa method [37], which search directly for the saddle point of the Lagrangian (2.11). In fact, if the Lagrangian $L(x, \lambda)$ admits a saddle point, say $(\hat{x}, \hat{\lambda})$, then the duality gap $\delta = \min_{x \in \mathcal{K}} \max_{\lambda \in \mathcal{R}^+} L(x, \lambda) - \max_{\lambda \in \mathcal{R}^+} \min_{x \in \mathcal{K}} L(x, \lambda)$ is null and \hat{x} is a critical point of the Lagrangian. Since the dual function $H(\lambda) = \operatorname{argmin}_{x \in \mathcal{K}} L(x, \lambda)$ is differentiable, it can be minimized explicitly by using the gradient descent method. Therefore the Uzawa method alternates a minimization step over \mathcal{R}^n with respect to x with λ fixed and a maximization step with respect to λ onto \mathcal{R}^+ , with x fixed. The algorithmic parameter $\rho > 0$ can be fixed to optimize convergence by relying on theoretical considerations. Therefore the CQP (2.9) is equivalent to finding

$$\hat{x} = \operatorname{argmin}_x \operatorname{argmax}_{\mu \geq 0} L(x, \mu) = \|x - y\|^2 + \langle \mu, Ax \rangle \quad (3.3)$$

3.1.3. Hildreth's algorithm

Hildreth [1] proposed to apply the Gauss Seidel algorithm [21] to the dual problem (2.9). A single cycle of the Hildreth's algorithm consists in updating each element of λ sequentially in an arbitrary fixed order. Therefore each cycle consists of m steps, each of which corresponds to a projection onto the cone $\mathcal{K}_i, i = 1, \dots, m$. The algorithm gives rise to a sequence of points, each one differing from the preceding in exactly one coordinate. At the cycle $k + 1$, the λ_i^{k+1} is used in the estimation of the point λ_{i+1}^{k+1} so that the best available estimations are used for computing each variable. The convergence of the Gauss Seidel algorithm is guaranteed only if the matrix A is full row rank, so that there are not redundancies among the inequality restrictions, and it is guaranteed independently of the initial point λ^0 only if A is positive definite and symmetric. The algorithm is sensitive to the normalization as well as to the order of the projections.

3.1.4. Primal-dual interior point methods

First introduced by Karmakar in 1984 [43], primal-dual interior point methods act by perturbing the complementarity condition $w^T \lambda = 0$ in the LCP formulation (2.15) and replacing with $w^T \lambda = \mu$. The partition of vectors w and λ into zero and nonzero elements is gradually revealed as the algorithm progresses by forcing a reduction of μ . All iterates satisfy the inequality constraints strictly. The solution is approached from either the interior or exterior of the feasible region but never lie on the boundary of this region. Let the function $F(x, \lambda, w)$ be such that the roots of this function are solutions to the first and the last optimality conditions in (2.15).

$$F_\mu(x, \lambda, w) = \begin{pmatrix} w - A^T x \\ x^T - y^T + \lambda^t A \\ w^T \lambda - \mu e \end{pmatrix}$$

The perturbed complementarity condition introduces a nonlinearity, therefore for each fixed $\mu > 0$ a system of nonlinear equations should be solved. The nonlinear system is typically solved by using a Newton-like algorithm [50]. Each iteration of the Newton's method finds a search direction from the current iterate (x_k, λ_k, w_k) . It is computationally expensive, but can make significant progress towards the solution. For instance in barrier methods, which are the most efficient of the family, this is achieved by using a penalizing term, called a barrier function, for violations of constraints whose value on a point increases to infinity as the point approaches the boundary of the feasible region. Interior point methods must be initialized at an interior point, or else the barrier function is undefined. The interested reader is referred to [44] for further information about interior point methods.

3.1.5. Dykstra's algorithm

In 1983, Dykstra [12] proposed a generalization of the Hildreth's procedure applicable to the case of constraints corresponding to more general convex cones than polyhedral convex ones. The Dykstra's algorithm is based on the idea, before suggested by Von Neumann [22] to the case of subspaces, of computing the projection onto the intersection of convex sets by relying on the solution of the simpler problem of projecting onto the individual sets. In the case of concave regression the projection onto a single convex set \mathcal{K}_i involves only three points and, if the constraint is not satisfied, it corresponds to the straight line fitting the points y_i, y_{i+1}, y_{i+2} .

Dykstra's algorithm iterates by passing sequentially over the individual sets and projects onto each one a deflected version of the previous iterate. More precisely, before projecting onto the cone \mathcal{K}_i during the $(k+1)$ -th cycle, the residuum obtained when projecting onto \mathcal{K}_i at the previous k -th cycle, say R_i^k is removed and a new residuum associated to the cone \mathcal{K}_i , say R_i^{k+1} is computed after the projection. In practice, each x_i^k is the projection of $y + R_1^k + \dots + R_{i-1}^k + R_{i+1}^k + \dots + R_m^k$ onto \mathcal{K}_i , where $R_i^k = x_i^k - (y + R_1^k + R_{i-1}^k + R_{i+1}^k + \dots + R_m^k)$.

If each \mathcal{K}_i is a subspace, then at each new cycle $k+1$ the residuum $-R_i^k$ of the projection onto each convex cone \mathcal{K}_i is of course the projection of x^k onto the cone \mathcal{K}_i^o . Therefore, the Dykstra's procedure for subspaces reduces to exactly the cyclic, iterated projections of von Neumann. In this case, for $k \rightarrow \infty$, the sum of the residua over the cones \mathcal{K}_i^k approximates the projection of y onto the polar cone \mathcal{K}^o and therefore, for the Moreau decomposition theorem, x^k approximates the projection onto \mathcal{K} .

However, if the \mathcal{K}_i are not subspaces, $\Pi(\cdot|\mathcal{K}_i)$ is not a linear operator and then the von Neumann algorithm does not necessarily converge.

The Dykstra's algorithm can also be interpreted as a variation of the Douglas-Rachford splitting method applied to the dual proximal formulation (2.23).

The seminal works of Hildreth and Dykstra have inspired many studies mostly devoted to theoretical investigations about their behavior in a Hilbert space [23, 24], about its convergence [25, 26], about its relation to other meth-

ods [27, 28] and about its interpretation in more general frameworks, such as the proximal splitting methods [29]. Han [30], as well as Iusem and De Pierro [25], showed that in the polyhedral case, the method of Dykstra becomes the Hildreth's algorithm and therefore it has the same geometric interpretation of Gauss Seidel to the dual problem. Gaffke and Mathar (1989) [27] showed the relation of the Dykstra algorithm to the method of component-wise cyclic minimization over a product space, also proposing a fully simultaneous Dykstra algorithm. The only works devoted to give some insight about a more efficient implementation are the ones of Ruud. Goldman and Ruud [9](1993) generalized the method of Hildreth showing that there is no need to restrict the iterations to one element of λ at a time. One can optimize over subsets or/and change the order in which the elements are taken. This observation is important for the speed of convergence since the slow speed can be understood as a symptom of near multicollinearity among restrictions. Because the intermediate projections are so close to one another, the algorithm makes small incremental steps towards the solution. They also remarked that Dykstra uses a parametrization in the primal parameter space, that causes numerical round off errors in the variable residuum. These round off errors cumulate so that the fitted value does not satisfy the constraints of the dual problem. It would be better to use a parametrization on the dual so that the constraints in the dual would be satisfied at each iteration. Later, Ruud [31] proved that the contraction property of the proposed generalizations rests solely on the requirement that every constraint appears in at least one subproblem of an iteration. As one approaches the solution, constraints that are satisfied at the solution are eliminated. To remove satisfied constraints would accelerate the Hildreth procedure. The authors propose to reduce the set of active constraints, that is the constraints satisfied as an equation at the corresponding points, by removing as many constraints as possible through periodic optimization over all positive elements of λ .

3.1.6. Alternating Direction Method of Multipliers (ADMM)

ADMM is an augmented Lagrangian technique [32, 33] which can be applied to problems of the form

$$\text{Find } \underset{z \in \mathcal{R}^m, Ax=z, z \leq 0}{\operatorname{argmin}} \|y - x\|^2 + g(Ax) \quad (3.4)$$

where the matrix A is assumed to be irreducible ($AA^T = vI, v > 0$) and the intersection of the relative interiors of the domains of the two functions is assumed to be not empty ($ri \operatorname{dom}(g) \cap ri \operatorname{dom}(f) \neq \emptyset$). ADMM minimizes the augmented Lagrangian \mathcal{L} over the two variables of the problems, say x and z , first x with z fixed, then over z with x fixed, and then applying a proximal maximization step with respect to the Lagrange multiplier λ . The augmented Lagrangian of index $\gamma \in [0, \infty]$ is

$$\mathcal{L}(x, z, y) = f(x) + g(z) + \frac{1}{\gamma} \lambda^T (Ax - z) + \frac{1}{2\gamma} \|Ax - z\|^2 \quad (3.5)$$

where $f(x) = \|y - x\|^2$. Denoting by $prox_f^A$ the proximal operator which maps a point $z \in \mathcal{R}^n$ to the unique minimizer of $f(x) + \|Ax - z\|^2$ and denoting by $prox_g = prox_{f \circ A}$ the implementation detailed in the Appendix B is obtained.

The ADMM method rests on the proximal formulation 2.21. Indeed, it can be viewed as an application of the Douglas-Rachford splitting algorithm [34].

3.2. Algorithms with finite-time convergence

All algorithms that will be reviewed in this section are *active set methods* resting on the LCP formulation 2.15. Active set methods work by choosing a subset of indices $j \in \tilde{J} \subset J = \{1, \dots, n\}$ such that w_j is allowed to be non-zero and forcing the corresponding λ_j to be zero, while the remaining indices $j \in J \setminus \tilde{J}$ force w_j to be zero and allow λ_j to take nonzero values. In this section we will review active set algorithm such as the mixed primal-dual basis algorithm (section 3.2.3), the critical index algorithm (section 3.2.4) and the Meyer's algorithm (section 3.2.5).

Before detailing the algorithms, we introduce some definitions and basic results about the geometry of polyhedral convex cones, on which are based the algorithms presented in this section. For further details the reader is referred to [38].

3.2.1. Properties of polyhedral convex cones with $m \leq n$

Lemma 1 establishes the relationship between the constraint matrix A and the edges of the polar cone $\{\gamma^i, i = 1, \dots, m\}$, namely $A^T = [\gamma^1, \dots, \gamma^m]$. We would like now to determine the edges of the constraint cone \mathcal{K} .

Let the vectors $\{\gamma^{m+1}, \dots, \gamma^n\}$ be orthogonal to the vectors $\{\gamma^i, i = 1, \dots, m\}$ and orthonormal to each other so that the set $\{\gamma^i, i = 1, \dots, n\}$ forms a basis for \mathcal{R}^n . By defining the dual basis of the basis $\{\gamma^i, i = 1, \dots, n\}$ as the set of vectors $\{\beta^i, i = 1, \dots, n\}$ that verify the relationship

$$(\beta^i)^T \gamma^j = \begin{cases} -1 & i = j \\ 0 & i \neq j \end{cases} \quad (3.6)$$

the constraint cone $\mathcal{K} = \{x : Ax \leq 0\}$ can be equivalently written as

$$\mathcal{K} = \left\{ x : x = \sum_{i=1}^m b_i \beta^i + \sum_{i=m+1}^n b_i \beta^i, b_i \geq 0, i = 1, \dots, m \right\} \quad (3.7)$$

To see that let $B = [\beta^1, \dots, \beta^n]$ and $C = [\gamma^1, \dots, \gamma^n]$. Then Ax are the first m coordinates of Cx . Since $B^T C = -I_n$ by construction, by multiplying both members at left for B^{-1} and at right for x , we obtain: $Cx = -B^{-1}x$. Therefore Cx gives the negative coordinates of x in the basis $\{\beta^i, i = 1, \dots, n\}$. Furthermore, points in \mathcal{K} have their first m coordinates non-negative and can be written as $x = \sum_{i=1}^n b_i \beta^i$, where $b_i \geq 0$ for $i = 1, \dots, m$.

Taking into account Def. 2, (3.7) established that the vectors β^i are the edges of the constraint cone \mathcal{K} .

Definition 4. Let \mathcal{K} be a polyhedral convex cone in \mathcal{R}^n , then $F \subseteq \mathcal{K}$ is a *face* of \mathcal{K} if and only if F is the intersection of \mathcal{K} with a supporting hyperplane.

A polyhedral convex cone arises as the intersection of a finite number of half-spaces whose defining hyperplanes pass through the origin. The i -th row of A is normal to the hyperplane generating the i -th closed half-space.

The following Lemma, proved by Rockafellar in 1970 [39], establishes a relationship between the first m vectors of the dual basis $\{\beta^i, i = 1, \dots, m\}$ and the faces of the cone $\mathcal{K}' = \mathcal{K} \cap \text{span}(\mathcal{K})$, where $\text{span}(\mathcal{K})$ denotes the subspace spanned by the m edges of \mathcal{K} .

Lemma 2. Let $\mathcal{K} = \{x : Ax \leq 0\}$, where $A^T = [\gamma_1, \dots, \gamma_m]$, be the constraint cone and let $\{\beta^i, i = 1, \dots, m\}$ be the dual basis of $\{\gamma^i, i = 1, \dots, m\}$. Denoting by $\text{span}(\mathcal{K})$ the subspace spanned by the m edges of \mathcal{K} , let $\mathcal{K}' = \mathcal{K} \cap \text{span}(\mathcal{K}) = \{x \in \mathcal{R}^n : x = \sum_{j=1}^m b_j \beta^j, b_j \geq 0\}$. Then, for $J \subseteq \{1, \dots, m\}$ the faces of \mathcal{K}' are the sets:

$$\left\{ x \in \mathcal{R}^n : x = \sum_{j \in J} b_j \beta^j, b_j \geq 0 \right\}$$

The set of all relatively open faces

$$\mathcal{F}_J = \left\{ x \in \mathcal{R}^n : x = \sum_{j \in J} b_j \beta^j, b_j > 0 \right\}$$

forms a partition of \mathcal{K}' .

Denoting by u and v the projections of y onto $\text{span}(\mathcal{K})$ and $\text{span}(\{\gamma^{m+1}, \dots, \gamma^n\})$ respectively, being the two subspaces orthogonal, y can be written as $y = u + v$. Since v can be easily computed as $v = X(X^T X)^{-1} X^T y$ with $X = [\gamma^{m+1}, \dots, \gamma^n]$, the problem (2.7) reduces to find the projection of u onto \mathcal{K}' .

The next Lemma, proved by Zarantonello [40] focuses on the set of points in $\text{span}(\mathcal{K})$ projecting on a given point $x \in \mathcal{K}'$, say $\Pi_{\mathcal{K}'}^{-1}(x)$. Before stating it we need to define the *support cone* of a closed convex set at a given point.

Definition 5. The support cone of a closed convex set \mathcal{K} at x denoted by $\mathcal{L}_{\mathcal{K}}(x)$ is the smallest convex cone with vertex at the origin containing $\mathcal{K} - x$.

Lemma 3. Let \mathcal{K}' , \mathcal{F}_J , $\{\gamma_i, i = 1, \dots, n\}$ and $\{\beta_i, i = 1, \dots, n\}$ defined as in Lemma 2. If x is a point of \mathcal{K}' belonging to the open face \mathcal{F}_J , then:

- $\Pi_{\mathcal{K}'}^{-1}(x) = x + \mathcal{L}_{\mathcal{K}'}^o(x) = \left\{ x + \sum_{i \notin J} c_i \gamma^i, c_i \geq 0 \right\} = \left\{ \sum_{j \in J} b_j \beta^j + \sum_{i \notin J} c_i \gamma^i, b_i > 0, c_i \geq 0 \right\}$,
where $x = \sum_{j \in J} b_j \beta^j$.
- The sets $\Pi_{\mathcal{K}'}^{-1}(x)$ are disjoint closed convex cones.
- $\cup_{x \in \mathcal{K}'} \Pi_{\mathcal{K}'}^{-1}(x) = \text{span}(\mathcal{K})$

where $\mathcal{L}_{\mathcal{K}'}^o(x)$ denotes the dual of the support cone of \mathcal{K}' at x .

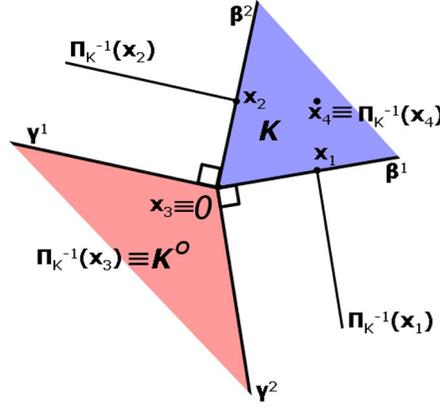


FIG 2. The point $x_1 \in \mathcal{K}$ belongs to the open face $F_J = \{x \in \mathcal{K} : x = b\beta^1, b > 0\}$, with $J = \{1\}$. The support cone of \mathcal{K} at x_1 is $\mathcal{L}_{\mathcal{K}}(x_1) = \{x : \gamma^{2T} x \leq 0\}$ and its dual is $\mathcal{L}_{\mathcal{K}}^o(x_1) = \{x : x = c\gamma^2, c \geq 0\}$. The set of points that project onto x_1 is given by the set $\Pi_{\mathcal{K}}^{-1}(x_1) = \{x_1 + \mathcal{L}_{\mathcal{K}}^o(x_1)\} = \{x_1 + c\gamma^2, c \geq 0\}$. The point $x_3 \in \mathcal{K}$ belongs to the open face $F_J = \{0\}$, with $J = \emptyset$. The support cone of \mathcal{K} at x_3 is \mathcal{K} and its dual is \mathcal{K}^o , so that the set of points that project onto x_3 is $\{\mathcal{K}^o\}$. The point $x_4 \in \mathcal{K}$ belongs to the open face $F_J = \{x \in \mathcal{K} : x = \sum_{i=1,2} b_i \beta^i, b_i > 0\}$, with $J = \{1, 2\}$. The support cone of \mathcal{K} at x_4 is the origin and its dual is the origin, so that the set of points that project onto x_4 is $\{x_4\}$.

Then any point in $\text{span}(\mathcal{K})$ projects onto a unique point in \mathcal{K}' and belong to a unique non-negative orthant, or sector S_J

$$S_J = \left\{ x \in \mathcal{R}^n : x = \sum_{j \in J} b_j \beta^j + \sum_{j \notin J} c_j \gamma^j, b_j > 0, c_j \geq 0 \right\} \quad (3.8)$$

Figure 2 illustrates this result for $\text{span}(\mathcal{K}) = \mathcal{R}^2$. Points in S_J project onto the subspace spanned by the vectors $\{\beta^j, j \in J\}$, that is on the face $F_J = \sum_{j \in J} b_j \beta^j$. Vectors belonging to \mathcal{K}^o project onto the origin, vectors belonging to \mathcal{K} project on themselves, while each other vector of \mathcal{R}^n project onto a unique face of \mathcal{K} .

Therefore, if the sector S_J containing the vector u is known, then the projection of u onto \mathcal{K} can be easily computed as the projection of u onto the subspace spanned by the $\{\beta^j, j \in J\}$. This reduces the problem of projecting y onto \mathcal{K} to the problem of finding the set of indices \hat{J} such that the sector $S_{\hat{J}}$ contains u . The set complementary of \hat{J} with respect to $\{1, \dots, m\}$ corresponds to the indices of the constraints satisfied at equality in the optimal solution.

The algorithms described in this section propose different strategies to find the optimal set \hat{J} .

3.2.2. Early algorithms based on the properties of polyhedral convex cones

The first algorithm addressing the problem of projecting a point $y \in \mathcal{R}^n$ onto a polyhedral convex cone $\mathcal{K} \subset \mathcal{R}^n$ by a non-asymptotic procedure dates back to

work of Wilhelmsen [13] in 1976. Wilhelmsen assumes that the m generators β^i of the cone $\mathcal{K} = \left\{x \in \mathcal{R}^n : x = \sum_{i=1}^m b_i \beta^i, b_i > 0\right\}$ are known and propose an algorithm which computes a sequence of nearest points x^k to y in subcones \mathcal{K}^k of \mathcal{K} . Each subcone \mathcal{K}^k is chosen so that $x^k \in \text{int}(\mathcal{K}^k)$ and is closer to y than is x^{k-1} . This means that x^k is in the near side of the supporting hyperplane of \mathcal{K}^{k-1} with respect to y . The key step is to find x^{k+1} given x^k and the proposed procedure to do that is laborious.

Pshenichny and Danilin (1978) [14] proposed an algorithm similar to the one of Wilhelmsen which also converges in a finite number of steps. In both algorithms, m can be any integer even larger than n . A more efficient procedure, but with the more restrictive assumption that $m \leq n$ has been proposed by Fraser and Massam in 1989.

3.2.3. Mixed primal-dual basis algorithm

Fraser and Massam [16] proposed an iterative algorithm to solve the general problem of projecting a data point $y \in \mathcal{R}^n$ onto a polyhedral convex cone $\mathcal{K} \subset \mathcal{R}^n$ generated by $m \leq n$ linear inequality constraints.

Polyhedral convex cones generated by a number of inequalities at least equal to the dimension of the space they belong to have been the subject of section 3.2.1. As seen there, the problem of projecting a data point onto this class of cones can be reduced to find the set of *edges*, or generators of the cone, indexed by $\hat{J} \subseteq 1, \dots, n$ such that the sector $S_{\hat{J}}$ contains the data point.

To this goal, the set of edges of the polar cone $\{\gamma^i, i = 1, \dots, m\}$ is completed by $n - m$ vectors orthogonal to $\{\gamma^i, i = 1, \dots, m\}$ and orthonormal to each other. In the case of concave regression $m = n - 2$, the set is completed by a constant function $\gamma^{m+1} = \mathbf{1}/\|\mathbf{1}\|$, where $\mathbf{1}$ is the m -dimensional unitary vector and by a linear function $\gamma^{m+2} = (x - \bar{x}\mathbf{1})/\|(x - \bar{x}\mathbf{1})\|$, where $x = (x_1, \dots, x_m)'$ and $\bar{x} = \sum_{i=1}^m x_i/m$. The set of vectors $\{\gamma^i, i = 1, \dots, n\}$ form a basis for \mathcal{R}^n .

Let the vectors $\{\beta_i, i = 1, \dots, n\}$ be the dual basis of the basis $\{\gamma^i, i = 1, \dots, n\}$ as defined in (3.6). Fraser and Massam called the vectors β^i and γ^i primal and dual vectors respectively. A primal-dual basis for \mathcal{R}^n , associated to the set of indices $J \subseteq \{1, \dots, n\} \equiv L$ is a basis $\mathcal{B}_J = [\alpha_1, \dots, \alpha_n]$ made up of a subset of the primal basis vectors $\{\beta_i\}_{i \in J}$ and a complementary subset of the dual basis vector $\{\gamma_i\}_{i \in L \setminus J}$. For $n = m$ the primal basis vectors, corresponding to the edges of \mathcal{K} , are simply the columns of $-(A^T)^{-1}$. Using the above definitions, the problem of projecting a point $y \in \mathcal{R}^n$ onto the cone \mathcal{K} can be formulated as follows.

Theorem 3.1. *The primal constrained quadratic minimization problem (2.7) is equivalent to the problem of finding*

$$\underset{x \in \mathcal{K}}{\operatorname{argmin}} \|u - x\|^2 \quad (3.9)$$

where $u = y - v$, with $v = \Pi(y|\operatorname{span}(\gamma^{m+1}, \dots, \gamma^n))$. Denoting by x_u the solution to this problem, the solution to the primal problem (2.7) is $\hat{x} = x_u + v$.

Finding the sector containing u is achieved moving along a fixed line joining an arbitrary chosen initial point x^0 inside the cone or on its boundary to the data point u . By moving along a fixed line, many sectors are crossed. Each time a sector is crossed the successive approximation x^k is obtained by projecting the point on the line passing through u and x^0 on the face F_{J^k} of \mathcal{K} (see Lemma 2) so that the distance $\|u - x^k\|$ is decreasing in k .

At each iteration each basis differs from another by one vector only and therefore the coordinates of x^k onto the new basis are easy to calculate. What changes is that one coordinate becomes equal to zero and therefore there is no need to estimate the inverse of a matrix at each step. For this reason this algorithm is faster than the one of Wilhelmsen.

Points belonging to the sector S_{J^k} have non-negative coordinates in the mixed primal-dual basis \mathcal{B}_{J^k} relative to the cone \mathcal{K} . Therefore the procedure terminates when the coordinates of the data point in the primal dual basis \mathcal{B}_{J^k} are all non-negative, meaning that the point x^k is on the face of the sector containing the data point u .

The number of iterations needed is equal to the number of different sectors that the line joining the initial point to the data point has to cross. This number is bounded above by 2^n . It is worth to remark that crossing a sector corresponds to a pivot step in the equivalent LCP. In fact, the net effect of a pivot step is of moving from the point x^k contained in the face F_{J^k} of \mathcal{K} to the point x_{k+1} contained in the face $F_{J^{k+1}}$ of \mathcal{K} .

Ten years later, Meyer [17] generalized the algorithm of Fraser and Massam to the case of more constraints than dimensions, that is when $m > n$.

3.2.4. Critical index algorithm: Nearest point problem in simplicial cones

Murty and Fathy (1982) [19] considered the general problem of projecting a given vector $y \in \mathcal{R}^n$ onto a *simplicial cone* $\mathcal{K} \subset \mathcal{R}^n$. The definition of simplicial cone and *pos cone*, on which the former definition is based are as follows.

Definition 6. The pos cone generated by the vectors in $\Delta = \{\delta^i, i = 1, \dots, m\}$, denoted by $pos(\Delta)$, is the set $\{x \in \mathcal{R}^n | x = \sum_{i=1}^m d_i \delta^i, d_i \geq 0\}$.

Definition 7. A cone $\mathcal{K} \subset \mathcal{R}^n$ is said to be *simplicial* if it can be expressed as a positive linear span of n linearly independent vectors $\Delta = \{\delta^i, i = 1, \dots, n\}$ in \mathcal{R}^n (i.e., a basis for \mathcal{R}^n): $\mathcal{K} = pos(\Delta)$.

For any point $x \in pos(\Delta)$ the vector $d = D^{-1}x$, where $D = [\delta^1, \dots, \delta^n]$ is called *combination vector* corresponding to x . Therefore the projection \hat{x} of y onto $pos(\Delta)$ can be expressed as a non-negative linear combination of the edges of the cone: $\hat{x} = \sum_{i=1}^n \hat{d}_i \delta^i$, where the optimal combination vector corresponding to \hat{x} is $\hat{d} = D^{-1}\hat{x}$.

Murty and Fathy named the set of indices $\hat{J} \subseteq \{1, \dots, n\}$ such that $\hat{d}_{i \in \hat{J}} > 0$ set of *critical indices*. Using the definitions of simplicial cone and combination vector, the original problem of projecting the point $y \in \mathcal{R}^n$ onto the cone \mathcal{K} can be formulated as follows.

Theorem 3.2. *The primal constrained quadratic minimization problem (2.7) is equivalent to the problem*

$$\hat{d} = \underset{d \geq 0}{\operatorname{argmin}} (u - Dd)^T W (u - Dd) \quad (3.10)$$

where $u = \Pi(y | \operatorname{span}(\mathcal{K}))$ and $D = [\gamma^1, \dots, \gamma^n]$, with $\{\gamma^i = A_i^T, i = 1, \dots, m\}$ and $\{\gamma^i, i = m + 1, \dots, n\}$ defined as in section 3.2.3.

Denoting by \hat{d} the solution to this problem, the solution to the primal problem (2.7) is $\hat{x} = y - D\hat{d} + v$.

This formulation has the same structure of the dual formulation (2.9), where the combination vector d in (3.10) plays the same role as the dual variable λ in (2.9). The only difference is that in (2.9) the matrix $A \in \mathcal{R}^{m \times n}$ is not squared as the matrix $D \in \mathcal{R}^{n \times n}$.

As shown in section 2.2 for the dual quadratic formulation (2.9), the formulation (3.10) can be recast as a LCP. This equivalency is important since the following theorem, proved in [19] applies to the LCP formulation of (3.10).

Theorem 3.3. *If a single critical index for the LCP problem of order n is known, the problem can be reduced to a LCP of order $n - 1$.*

The fact that finding a critical index reduces the dimension of the problem can be argued geometrically. Let l be a critical index, then denoting by $NPP[\Gamma; u]$ the subproblem (3.10), where $\Gamma = \{\gamma^i, i = 1, \dots, n\}$ its solution is also the solution to the $NPP[\Gamma \cup \{-\gamma^l\}; u]$. Defining $\bar{u} = u - \frac{\gamma^l (u^T \gamma^l)}{\|\gamma^l\|^2}$ and $\bar{\Gamma} = \{\bar{\gamma}^1, \dots, \bar{\gamma}^{l-1}, \bar{\gamma}^{l+1}, \dots, \bar{\gamma}^n\}$, where $\bar{\gamma}^i = \gamma^i - \frac{\gamma^l (u^T \gamma^i)}{\|\gamma^l\|^2}$, then $\bar{\gamma}^i, i \in \{1, \dots, n\} \setminus l$ is orthogonal to γ^l and the cone $\operatorname{pos}(\Gamma \cup \{-\gamma^l\})$ is the direct sum of the full line generated by γ^l and the simplicial cone $\operatorname{pos}(\bar{\Gamma})$.

Solving $NPP[\bar{\Gamma}, \bar{u}]$ is an $n - 1$ dimensional problem. If \hat{x}^* is the solution of the $NPP(\bar{\Gamma}, \bar{u})$, then the solution \hat{x} to the $NPP[\Gamma; u]$ is obtained as $\hat{x} = \hat{x}^* + \frac{\gamma^l (u^T \gamma^l)}{\|\gamma^l\|^2}$.

By relying on Theorem 3.3, the authors proposed an algorithm consisting of a subroutine to identify a critical index for the problem, followed by a subroutine which reduces the size of the problem once a critical index is found. Since the solution \hat{d} is the orthogonal projection onto the linear hull of $\{\gamma^i, i \in \hat{J}\}$, if \hat{J} is known, the solution of the equivalent $LCP(q, M)$ and correspondingly the solution to $NPP[\Gamma; u]$ can be easily found.

The routine to identify a critical index operates on the $NPP[\Gamma; u]$ by exploiting the geometric properties of projection faces of a pos cone, whose definition is as follows.

Definition 8. Let $S \subset \Gamma$. $\operatorname{pos}(S)$ is a face of $\operatorname{pos}(\Gamma)$ of dimension $|S|$. $\operatorname{pos}(S)$ is said to be a projection face of $\operatorname{pos}(\Gamma)$ if $\Pi(u | \operatorname{span}(S)) \in \operatorname{pos}(S)$.

In the following, we list some theorems on which the critical index algorithm is based. Their proofs can be found in [41] (chapter 7).

Theorem 3.4. *Let $S \subset \Gamma$, $S \neq 0$. The optimum solution of $NPP[S; u]$ is in the relative interior of $\text{pos}(\Gamma)$ if and only if the projection of u onto the linear span of S is in the relative interior of $\text{pos}(S)$: $\Pi(u|\text{span}(S)) \in \text{ri}(\text{pos}(S))$.*

Theorem 3.5. *Let $\hat{x} = D\hat{d}$ be the optimum solution of $NPP[\Gamma; u]$. Let \hat{J} the set of critical indices and $S = \{\gamma^j, j \in \hat{J}\}$. Then $\text{pos}(S)$ is a projection face of $\text{pos}(\Gamma)$.*

Theorem 3.5 tells us that the projection onto the cone \mathcal{K}^o belongs to the pos cone generated by the set S of vectors corresponding to critical indices and that such pos cone is a projection face. Therefore, if the set of critical indices is known, for Theorem 3.4 the solution can be computed as the projection onto the linear subspace spanned by the vectors in S .

The routine maintains a non-empty subset of Γ called the *current set* denoted by S , and a point called the *current point* denoted by \bar{x} . At each stage of the routine $\bar{x} \in \text{pos}(S)$. When termination occurs the routine either finds the nearest point in $\text{pos}(\Gamma)$ to u in which case the problem is completely solved or it finds a critical index of the problem. In the latter case, an LCP of order $(n - 1)$ can be constructed and the same routine can be applied to this smaller problem. Hence, the unique solution of the original problem can be obtained after at most n applications of the routine which finds a critical index.

A characterization useful to find a critical index or the solution to the problem is provided by the following theorem.

Theorem 3.6. *Let $\bar{x} \in \text{pos}(\Gamma)$ be such that $0 \in T(u, \bar{x})$, where $T(u, \bar{x})$ is the tangent hyperplane at \bar{x} to the ball of center u and radius \bar{x} . If there exists an index j such that $(u - \bar{x})^T \gamma^i \leq 0$ for all $i \neq j$ and $(u - \bar{x})^T \gamma^j > 0$ then j is a critical index of $NPP(\Gamma, u)$.*

A characterization of the optimal solution in terms of separating hyperplanes is given by Robertson et al. [42].

Theorem 3.7. *A point $\bar{x} \in \text{pos}(\Gamma)$ is the nearest point in $\text{pos}(\Gamma)$ to y if and only if $0 \in T(y; \bar{x})$ and $(y - \bar{x})^T \gamma^j \leq 0$, $\forall j = 1, \dots, n$, where $T(y, \hat{x})$ is the tangent hyperplane to $\text{pos}(\Gamma)$ in \hat{x} .*

The routine to find a critical index alternates distance reduction operations with line-search and projection steps to find a projection face. In practice, the routine starts by projecting on the closest edge to the data point. If the optimality condition is not satisfied, then the procedure iteratively adds vectors to S and updates the point \bar{x} while consistently reducing the distance between u and \bar{x} . The distance reduction operation is carried out efficiently by projecting onto the non-negative hull of two vectors in \mathcal{R}^n , the current point \bar{x} and a vector γ^i satisfying one of the conditions given by the following theorem.

Theorem 3.8. *Given $\bar{x} \in \text{pos}(\Gamma)$, $\bar{x} \neq 0$ such that $0 \in T(y, \bar{x})$, if for some $i \in \{1, \dots, n\}$ we have $(y - \bar{x})^T \gamma^i > 0$ and either:*

- $\|\bar{x} - y\| \leq \|\Pi(y|\gamma^i) - y\|$ and $\{\bar{x}, \gamma^i\}$ is linearly independent, or
- $y^T \gamma^i \leq 0$

then the projection of y onto the linear hull of $\{\bar{x}, \gamma^i\}$ is in the relative interior of $\text{pos}\{\bar{x}, \gamma^i\}$

Once such updates are no longer possible, it employs a sequence of line-search steps and projections in the subspace spanned by the vectors in S to find a projection face of the corresponding pos cone. This line-search is in the same spirit as the one proposed by Fraser and Massam since the goal is to reduce the distance to the data point while keeping at the interior of a pos cone. In the particular case of concave regression, for which $m < n$, it can be implemented exactly in the same way.

This algorithm is much faster than the MPDB algorithm. The primary source of its computational efficiency is that it relies mostly on distance reduction operations and size reduction steps whose computational requirement is relatively small compared to the computational effort required to find a projection face through a line-search.

Recently, Liu and Fathy (2011) [20] generalized the work of Murty and Fathy (1982) to polyhedral non-simplicial cones, hence allowing the set Γ to contain more than n vectors. What allows the generalization is the equivalence between the structure of the two problems through the concept of a polar cone.

The authors also proposed several strategies for efficient implementation mostly based on the mathematical properties of the entities involved. We incorporated, where possible, these strategies, to all algorithms we evaluated.

3.2.5. Meyer's algorithm

Meyer [18] considered the general problem of projecting a data point $y \in \mathcal{R}^n$ onto a polyhedral convex cone $\mathcal{K} \subset \mathcal{R}^n$ generated by a finite number m of linear inequalities. The problem is reduced to find the set of indices $\hat{J} \subset \{1, \dots, M\} \equiv L$, where $M \leq m$ is the number of linearly independent constraints, corresponding to not saturated constraints at the solution. Meyer called these indices *hinges*.

When $m \leq n$, the algorithm can be applied to both the primal and the dual formulation, whereas for $m > n$ it is applied to the dual formulation. In the following, since for the problem of concave regression $m < n$, we consider how to solve the primal problem (2.7).

The search for \hat{J} is performed iteratively, starting with an initial guess by removing or adding one index at time, until the optimal solution is obtained. For each candidate J^k two conditions are tested: the *interior point condition* and *optimality condition*.

The interior point condition is satisfied when the current iterate belongs to the interior of $\text{pos}(S)$, that is when $x^k \in \text{pos}(S)$, where $S \subset \{\beta^i, i \in J^k\}$. By using the following theorem

Theorem 3.9. *Let $S \subset \{\beta^i, i = 1, \dots, m\}$, $S \neq \emptyset$. The optimum solution of problem (2.7) is in the relative interior of $\text{pos}(S)$ if and only if $\Pi(y|\text{span}(S))$ is in the relative interior of $\text{pos}(S)$.*

x^k can be computed as projection of y onto the linear hull spanned by the vectors $\{\beta^i, i \in J^k\}$. If the feasibility condition is not satisfied, the index $j \in L \setminus J^k$ corresponding to the most negative coefficient is added to J^k and the interior point condition is checked again.

Once the feasibility condition is satisfied, the optimality condition is tested by using the characterization given in Theorem 3.7. If it is not satisfied, the vector $\beta^i, i \in J^k$ which most violates the condition is removed. The procedure continues until both conditions are satisfied.

Convergence is guaranteed by the fact that when the algorithm replaces just one edge, the Sum of the Squared Errors (SSE) after is less than the SSE before so that the algorithm never produces the same set of edges twice, which would result in an infinite loop.

In practice, each time a hinge is added, the best solution with $n + 1$ hinges where the first n hinges are already given is obtained. But this is not in general the best fit with $n + 1$ hinges, so that some hinge may need to be changed. Therefore, the optimal solution can be interpreted as the best approximation with the biggest possible number of hinges.

4. Issues about effectiveness for large-scale problems

In this section, we discuss strengths and limitations of the algorithms detailed above in solving large-scale instances of a particular kind of cone regression, the concave regression. In particular, we consider computational issues related to numerical stability, computational cost and memory load as well as the suitability to take advantage of a good initialization and to be implemented in an online fashion.

4.1. Suitability to take advantage of available good estimates

One general strategy for reducing the computational cost of a large-scale optimization problem is to use an initial guess that is easier to calculate and close to the optimal solution.

Within the class of algorithms with asymptotic convergence, splitting-based methods work by activating each of the convex constraints repetitively and by combining them to obtain a sequence converging to a feasible point. Since the projection point $\Pi(y|\mathcal{K})$ is characterized by the variational inequality

$$\hat{x} = \Pi(y|\mathcal{K}) \in \mathcal{K}, \quad \forall x \in \mathcal{K} : \langle y - \hat{x}, x - \hat{x} \rangle \leq 0 \quad (4.1)$$

the projection operator $\Pi(\cdot|\mathcal{K})$ is a closed contraction. Therefore the set of fixed points of $\Pi(\cdot|\mathcal{K})$ is exactly \mathcal{K} . This prevents the use of an initialization point belonging to the feasible set as well as the use of multiscale strategies since there is no guarantee that the solution from a previous level does not belong to the feasible set.

The same difficulty arises when considering interior point algorithms, since they need to be initialized to an interior point. In [9], Goldman proved that the

Dijkstra's algorithm can potentially start from better starting values than the given data point y . The author established the convergence to the nearest point to the primal cone from an arbitrary point in the intersection of \mathcal{C} and the ball of radius $\|y\|$, where $\mathcal{C} = \{x | x = y - A^T \lambda, \lambda \geq 0\}$, is a rotation of π radians of the dual cone \mathcal{K}^o with its vertex translated at y . A point satisfying these conditions can be obtained efficiently by using distance reduction operations based on Theorem 3.8. It is worthwhile to remark that this result can be easily interpreted in the active set framework. In fact, the Dykstra's algorithm can be understood as a primal active set method and its solution is a primal dual feasible point. Therefore, any dual feasible point, that is every point belonging to the set \mathcal{C} , can be used as initialization.

All algorithms with time finite convergence detailed in the previous section are primal-dual and they reduce the problem of projecting a given data point onto a convex set to the problem of finding the set of indices corresponding to not saturated constraints at the solution. In general, they involve the selection of a subset from a collection of items, say $\hat{J} \subseteq \{1, \dots, m\}$. With this formulation, they potentially allow us to take advantage of a good estimate of the optimal active set. However, the adaptation is not rapid since the active set estimate is updated using one-by-one changes preventing this class of method from being effective general-purpose solvers for large-scale problems.

In the algorithm of Fraser and Massam, the set of successive approximations are obtained by moving along a fixed line connecting the initial guess to the data point. The number of iterations needed to reduce the data point reduces the number of sectors to be crossed to join the sector containing the data point.

By contrast, in the algorithm of Meyer the proximity of the initial guess to the data point is not a good criterion selection for the initial guess. In fact, given an initial guess, the solution is attained by adding and/or removing one index at time until an optimal solution is found. Taking into account that the optimal solution can be interpreted as the best approximation with the biggest possible number of hinges, if the optimal solution contains just a few hinges, than using the empty set as an initial guess would be much faster than using the full set of possible hinges. On the other hand, if just a few constraints are satisfied at equality in the optimal solution, than the full set of indices will be a much better initial guess than the empty set. Therefore even if the choice of the initial guess may highly influence the performances, its choice depends on the data and there is not a well established criterion to fix it.

The Murty and Fathy's algorithm reduces the size of the problem each time a critical index is found. Therefore it is not compatible with strategies that take advantage of a good initial estimate since a good estimate does not lead to finding a critical index faster.

To overcome the limitation of active set methods in taking advantage of a good estimate, Curtis et al. [45] have recently proposed an heuristic framework that allows for multiple simultaneous changes in the active-set estimate, which often leads to a rapid identification of the optimal set. However, there is no guarantee of the computational advantages for general problems and, furthermore, the authors recommend their approach when solving generic quadratic

programming problems with many degrees of freedom. That is not the case of general concave regression problems.

4.1.1. PAV's inspired approximate solution

To evaluate through numerical simulations the suitability of taking advantage of good initial estimates, we propose an algorithm inspired to Pool Adjacent Violators (PAV), whose computational complexity is $\mathcal{O}(n)$. Starting from the original signal, violated constraints are removed one by one by projecting the current iterate onto the convex cone corresponding to the violated constraint until a primal feasible solution is obtained. Since the dual feasibility of each iterate is not guaranteed, the founded solution is not optimal. However, in our experience the solution is a very good approximation of the optimal solution.

4.2. Suitability to be implemented in an online fashion

Another strategy to deal with large projection problems would be to build and evaluate the solution incrementally according to the order of its input, as done in online methodologies developed for dynamic optimization [46] over a stream of input. Even if the input is given in advance, inputs are processed sequentially and the algorithm must respond in real-time with no knowledge of future input. Each new input may cause rising or falling constraints and the final solution is a sequence of feasible solutions, one for each time step, such that later solutions build on earlier solutions incrementally.

Of course this strategy requires that the algorithm responds in real-time for each new input. That would not possible when dealing with large matrix inverse computations. Let $\hat{x} \in \mathcal{R}^n$ be the projection of $y \in \mathcal{K}^n$ onto \mathcal{K} and let $\hat{x} \in \mathcal{K}^{n+1}$ be the projection of $\bar{y} \in \mathcal{R}^{n+1}$ onto \mathcal{K}^{n+1} . When a new element is added to the data point, a new constraint is added too, so that the constraint cone has a new edge. If this constraint corresponds to a critical index, that is to a constraint satisfied at equality in the optimal solution, then the projection face will be the same so that no further computing will be needed. On the contrary, if the new constraint does not correspond to a critical index, the projection face will change, including the edge corresponding to the new constraint and removing and adding some others. Therefore, the major difficulty faced by an online strategy is the same faced in exploiting good estimates.

4.3. Computational issues

As highlighted in the previous section, despite the different strategies implemented by algorithms with finite-time convergence, generally an index at time is iteratively added or removed from the current set of indices J^k until both the feasibility condition and the optimality condition are satisfied. Checking the optimality condition involves computing the inverse of a matrix that differs

slightly from the matrix of the previous iteration. What 'slightly' exactly means depends on the specific algorithm and it is detailed in the following.

The algorithm of Fraser and Massam involves the computation of an $n \times n$ fixed size matrix inverse at each iteration. The matrix to be inverted differs from the matrix of the previous iteration only for one column, being the change of the form $A \rightarrow (A + u \times v)$, where u is a unitary vector with only one nonzero component corresponding to the index of the column to be changed, and v corresponds to the difference between the elements of the dual basis vector and the elements of the primal basis vector or vice versa. In this case the inverse can be updated efficiently by using the Sherman-Morrison formula: $(A + u \times v)^{-1} = \frac{z \times w}{1 + \lambda}$, where $z = A^{-1}u$, $w = (A^{-1})^T v$, $\lambda = v^T z$. Therefore only two matrix multiplications and a scalar product are needed to compute the new inverse at each step.

The algorithm of Meyer, as well as the one of Liu and Fathi [20] involve the computation of an inverse matrix of variable size at each iteration. The matrix to be inverted differs from the matrix of the previous iteration only for one column, which has been added or removed. Since the matrix to be inverted $A(J) \in \mathcal{R}^{r \times n}$, with $r \leq m$ is generally rectangular, the Moore-Penrose generalized inverse or pseudoinverse is computed: $A(J)^\dagger = A(J)^T (A(J)A(J)^T)^{-1}$.

In Matlab and Scilab, the computation of the pseudoinverse is based on the Singular Value Decomposition (SVD) and singular values lower than a tolerance are treated as zero. The advantage of this approach is that the pseudoinverse can be computed also for a nonsingular matrix. However, the method proposed by Liu and Fathi [20] to improve the computational efficiency of their algorithm does not take advantage of the SVD approach since it consists in updating the matrix $(A(J)A(J)^T)^{-1}$. If the matrix $A(J)$ is ill-conditioned, then the inverse cannot be computed with good accuracy and for the matrix $A(J)A(J)^T$ is even more so because this operation squares the condition number of the matrix $A(J)$.

A better solution would be to update directly the pseudoinverse. This can be achieved when a column is added by using the method proposed in [47] Let $A^T \in \mathcal{R}^{n \times r}$, $x \in \mathcal{R}^n$ and $B = \begin{pmatrix} A^T & x \end{pmatrix} \in \mathcal{R}^{n \times (r+1)}$. The pseudoinverse of B can be computed from the pseudoinverse of A^T as follows.

$$B^\dagger = \begin{pmatrix} A^\dagger - A^\dagger x w^\dagger \\ w^\dagger \end{pmatrix}, \text{ where } w = (I - AA^\dagger)x \text{ and } w^\dagger = \frac{w^T}{\|w\|^2}.$$

Alternatively, the transformation $A(J)^T (A(J)A(J)^T)^{-1} y$ can be efficiently computed by a QR decomposition approach. Let $A^T = (Q_{11} Q_{12}) \begin{pmatrix} R_{11} \\ 0 \end{pmatrix}$ be the QR decomposition of A^T , where R_{11} is an $m \times m$ invertible upper triangular matrix. Then: $A(J)^T (A(J)A(J)^T)^{-1} = Q_{11} (R_{11}^T)^{-1}$. The inverse of an upper triangular matrix can be efficiently implemented by a left matrix division or by more sophisticated methods as the one proposed in [48].

Courrieu [49] proposed a method based on the full rank Cholesky decomposition which has a computation time substantially shorter of the method based on

SVD decomposition. The two main operations on which his method is based are the full rank Cholesky factorization of $A^T A$ and the inversion of $L^T L$, where L is a full rank matrix. On a serial processor these computations are of complexity order $\mathcal{O}(n^3)$ and $\mathcal{O}(m^3)$ respectively. However, in a parallel architecture, with as many processors as necessary, the time complexity for Cholesky factorization of $A^T A$ could reduce to $\mathcal{O}(n)$, while the time complexity for the inversion of the symmetric positive definite matrix $L^T L$ could reduce to $\mathcal{O}(\log(r))$. However, the computational advantage of this method can be appreciated only when $r \ll n$, since the inverse of a $r \times r$ matrix has to be computed, which is not in general the case, specially for concave regression problems.

The method proposed by Zhu and Li [51] for recursive constrained least squares problems, found that the exact solution of Linear Equality-constrained Least Squares can be obtained by the same recursion as for the unconstrained problem, provided that the Restricted Least Squares procedure is appropriately initialized. However, even this approach offers significant advantages in term of computational cost only when the number of constraints m is small, which is not the case for large-scale concave regression problems.

5. Improving the active set framework for concave regression problems

An active set algorithm for solving the concave regression problem generates a sequence of quasi stationary points. A primal (dual) feasible active set algorithm generates a sequence of primal (dual) feasible quasi stationary points with decreasing objective function values and terminates when the dual (primal) feasibility is satisfied. An active set J induces a unique partition of the indices $\{1, \dots, n\}$ into blocks. A block B of such partition is a set of consecutive integers, say, $\{p, p+1, \dots, q\}$, such that the index i of the constraint $x_{i+1} - x_i \leq x_{i+2} - x_{i+1}$ is in J for each i such that $p \leq i \leq q-2$. Conversely any such partition of indices determines a unique active set. Denoting by λ_i the multiplier associated with the i th constraint, the KKT optimality conditions can be written as:

$$\begin{cases} x_{i+1} - x_i \leq x_{i+2} - x_{i+1} & i = 1, \dots, n-2 \\ y_1 - x_1 = \lambda_1 \\ y_2 - x_2 = -2\lambda_1 + \lambda_2 \\ y_3 - x_3 = \lambda_1 - 2\lambda_2 + \lambda_3 \\ y_{n-1} - x_{n-1} = \lambda_{n-4} - 2\lambda_{n-3} + \lambda_{n-2} \\ y_n - x_n = \lambda_{n-3} - 2\lambda_{n-2} \\ y_n - x_n = \lambda_{n-2} \\ \lambda_i \geq 0 & i = 1, \dots, n-2 \\ \lambda_i(2x_{i+1} - x_i - x_{i+2}) = 0 & i = 1, \dots, n-2 \end{cases}$$

It is easy to show that: $\sum_{i=1}^n i(y_i - x_i) = 0$ and $\sum_{i=1}^n (y_i - x_i) = 0$. Knowing that each block can be represented by an affine function $x_i = \alpha + \beta i$, the above

system can be written as:

$$\begin{cases} \sum_{i=1}^{n_1} y_i = \alpha^1 \sum_{i=1}^{n_1} i + \beta^1 n_1 + \lambda_{n_1} \\ \sum_{i=n_1+1}^n y_i = \alpha^2 \sum_{i=n_1+1}^n (i - n_1) + \beta^2 (n - n_1) - \lambda_{n_1} \\ \sum_{i=n_1+1}^n y_i = \alpha^2 \sum_{i=n_1+1}^n (i - n_1) + \beta^2 (n - n_1) - \lambda_{n_1} \\ \sum_{i=n_1+1}^n (i - n_i) y_i = \alpha^2 \sum_{i=n_1+1}^n (i - n_1)^2 + \beta^2 \sum_{i=n_1+1}^n (i - n_1) - \lambda_{n_1} \\ \lambda^1 n_1 + \beta^1 - \alpha^2 - \beta^2 = 0 \end{cases}$$

Therefore, for each block the unknown variables to be computed are α, β, λ . The systems to be solved can be written as $A\mathbf{x} = \mathbf{b}$, where $\mathbf{x} = (\alpha^1, \beta^1, \alpha^2, \beta^2, \lambda_{n_1})$, $\mathbf{b} = (\sum_{i=1}^{n_1} y_i, \sum_{i=n_1+1}^n y_i, \sum_{i=n_1+1}^n y_i, \sum_{i=n_1+1}^n (i - n_i) y_i, 0)$ and

$$A = \begin{pmatrix} \sum_{i=1}^{n_1} i & n_1 & 0 & 0 & 1 \\ 0 & 0 & \sum_{i=1}^{n-n_1} i & n - n_1 & -1 \\ \sum_{i=1}^{n_1} i^2 & \sum_{i=1}^{n_1} i & 0 & 0 & n_1 \\ 0 & 0 & \sum_{i=1}^{n_2-n_1} i^2 & \sum_{i=1}^{n_2-n_1} i & 0 \\ n_1 + 1 & 1 & -1 & -1 & 0 \end{pmatrix}$$

In general, if k is the number of blocks, than the system to be solved has size $3k - 1$. As observed in [52], in the case of concave regression the number of blocks at the solution is usually much lower than n . Therefore, an active set algorithm that starts with an empty active set, should found the solution without the need of inverting large matrices.

6. Comparative results

In Table 1 we compare qualitatively the algorithms analyzed above in terms of their formulation (primal, dual, or primal-dual), their suitability to take advantage of a good initialization and their major limitations in dealing with large scale problems. All analyzed methods are dual or primal-dual. Dual methods cannot be initialized, whereas initialization in primal-dual methods is allowed but constrained. The major limitation of asymptotic convergence methods when dealing with large problems is the convergence, whereas finite-time convergence methods become too slow and numerically unstable because of accumulation errors.

In the following, we provide evidence of these limitations through numerical simulations reported in the Appendix A and we compare quantitatively their performances in term of distance from the solution for one second, one minute and ten minutes of CPU time. Instead of using only unstructured random data for testing, we considered signals of increasing difficulty level varying from a noisy concave signal to a concave/convex noisy signal. More precisely, we considered three signals, whose equations and plots are given in Figure 3 of the Appendix A and, for each of them, we considered three different sizes: $n \in \{50, 500, 1000\}$ and three increasing levels of noise (standard deviations $\sigma \in \{0.01, 0.1, 0.5\}$). To evaluate robustness against initialization varia-

TABLE 1
Qualitative comparison of all algorithms

Algorithm	Type	Formulation	Initialization	Limitation
Hildreth	dual	2.9	not possible	convergence
Dykstra	primal-dual	2.7	constrained	convergence
ADMM	dual	3.4	not possible	convergence
LSPS	primal	3.1	constrained	convergence
Uzawa	dual	3.3	not possible	convergence
MPDB	primal-dual	3.9	constrained	slow or unstable
Meyer	primal-dual	2.7	constrained and not robust	slow or unstable
Critical Index	dual	3.10	not possible	slow or unstable

tion for finite-time active set methods, we considered three different initializations of the active set J : the empty set, the set of m indexes corresponding to the linear inequality constraints, and the set of not saturated constraints obtained by using the algorithm inspired to PAV described in the previous section.

In the class of algorithms with asymptotic convergence, the most efficient appears to be ADMM. This is evident even for very small size data, when using difficult signals such as near-convex signals or very noisy signals (see Figure 5 and Figure 6 of the Appendix A). For noisy concave signals (Figure 4 of the Appendix A), the computational efficiency of ADMM is more evident in the presence of a high level of noise. Already for signals of size 500 the performance of ADMM is not very good; convergence ($SSE < 10^{-6}$) is not completely attained. The algorithm of Meyer is very sensitive to the initialization. It gives good performances when the signal is a Gaussian white noise and the initial active set is empty since most of constraints are expected to be saturated at the solution (see Figure 6 and Figure 9 of the Appendix A). Given a good initialization, the MPDB algorithm allows to compute the exact solution faster than other methods. However, this algorithm is not numerically stable since it requires one to compute the inverse of a matrix at each iteration. As explained in section 4.3, this is done incrementally by using the Sherman-Morrison formula. However, numerical round-off errors cumulate so that, in our implementation, the exact inverse is computed each 150 iterations. As it can be observed in Figure 10, Figure 11 and Figure 12 of the Appendix A, that refer to signals of size 1000, sometimes the round-off error dominates the calculation and the distance to the solution increases instead of decreasing.

These results demonstrated that, although there have been theoretical and practical advances in recent years, the use of shape-constrained nonparametric techniques for solving large scale problems (more than many thousand of points) is still limited by computational issues. To deal with very large scale problems, up to a million of points, matrix inverse calculation should be avoided and more efforts should be devoted to find a way to better initialize primal-dual methods, by computing approximate solutions and exploiting multi-scale strategies.

7. Conclusions

In this paper we have stated, analyzed and compared qualitatively and quantitatively several optimization approaches for solving the cone regression problem. We have distinguished two broad classes of methods. On the one side, methods with asymptotic convergence that rest on the properties of proximity operators and act by finding the solution as the limit of a sequence of successive approximations. On the other side, methods with finite-time convergence that exploit the geometric properties of polyhedral convex cones and find the exact solution as non-negative linear combination of functions, forming a basis in a specified finite dimensional space.

Simulations up to one thousand of points have demonstrated that the choice of the optimization approach severely impacts algorithmic performances. In particular, it has emerged that methods with finite-time convergence are much more efficient with respect to asymptotic-convergence methods. However, from this study it emerged that they face a twofold difficulty to cope with large-scale optimization. The first difficulty arises from the fact that all algorithms of this class modify the active set estimate one-by-one, making the adaptation of the initial active set estimation very slow; the second difficulty lies in the fact that they involve the computation of the inverse or the pseudoinverse of a matrix at each variation of the active set. Although there exists many methods to do that efficiently when the matrix rank is much lower than the size of the data, this condition cannot be assured in general. Incremental strategies to reduce the cost of computing the inverse of a matrix when the inverse of a slightly different matrix is known, are bounded by round-off error accumulation in an iterative setting.

The results of this study suggest that to be able to treat very large-scale problems (up to a million of points) further research should focus on finding a way to exploit classical multi-scale strategies and to compute an approximate solution through a penalization or splitting method without involving any matrix inverse calculation.

Appendix A

A.1. Signals used for comparative evaluations

$$S_1(z) = \begin{cases} 2n \sin(\frac{24}{5n}z) & z = 1, \dots, \frac{n}{3} \\ \alpha + \beta z & z = \frac{n}{3} + 1, \dots, \frac{2n}{3} \\ \gamma z^3 + \delta & z = \frac{2n}{3} + 1, \dots, n \end{cases}$$

$$\text{where } \beta = \frac{1}{10}, \alpha = 2n \sin(\frac{8}{5}) - \beta n, \gamma = \frac{-2}{n^2} \text{ and } \delta = \alpha + \beta \frac{2n}{3} - \gamma \frac{8n^3}{27}.$$

$$S_2(x) = \mathcal{N}(\mu, \sigma^2)$$

$$S_3(z) = \text{sinc}(\frac{6}{n}z - 1)$$

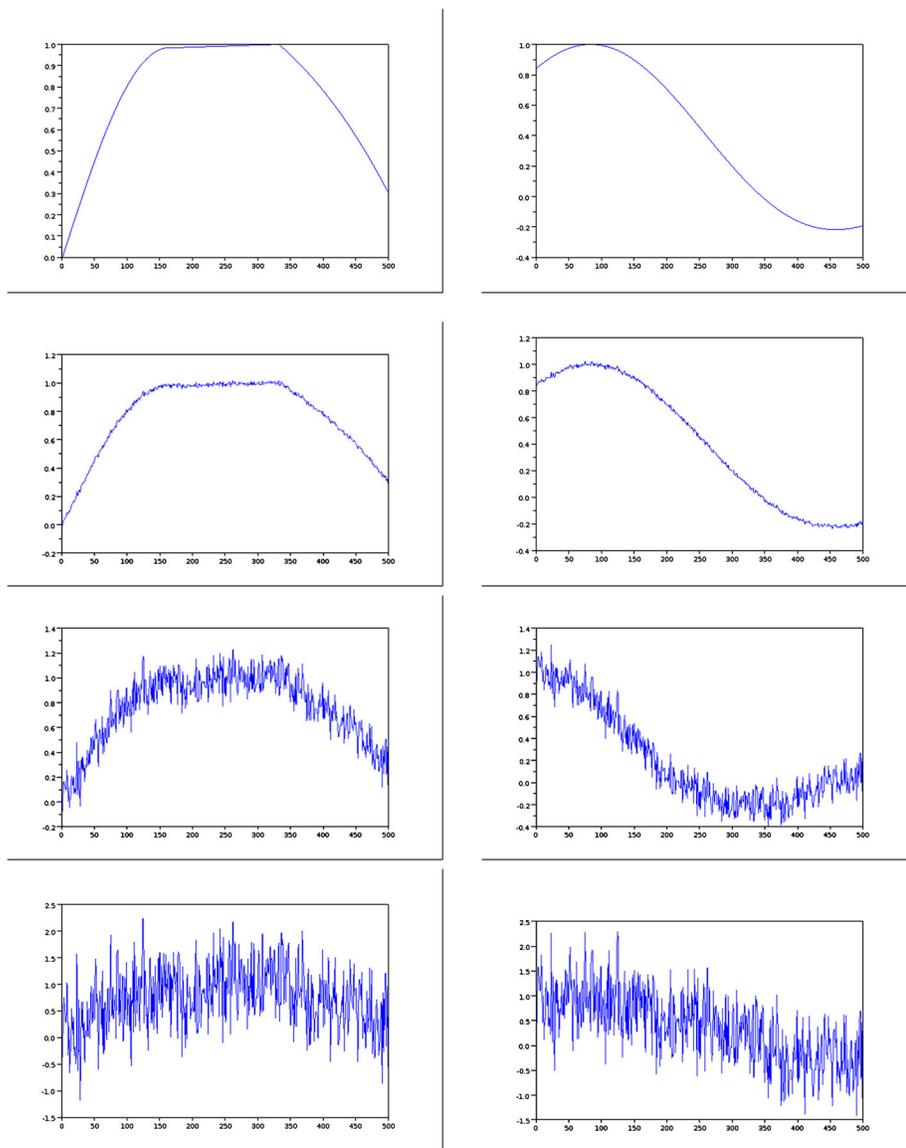


FIG 3. Signals S_1 (top left) and S_2 (top right) to which has been added white noise with three different values of standard deviation $\sigma = 0.01$, $\sigma = 0.1$, and $\sigma = 0.5$.

A.2. Comparative evaluations on signals of size 50

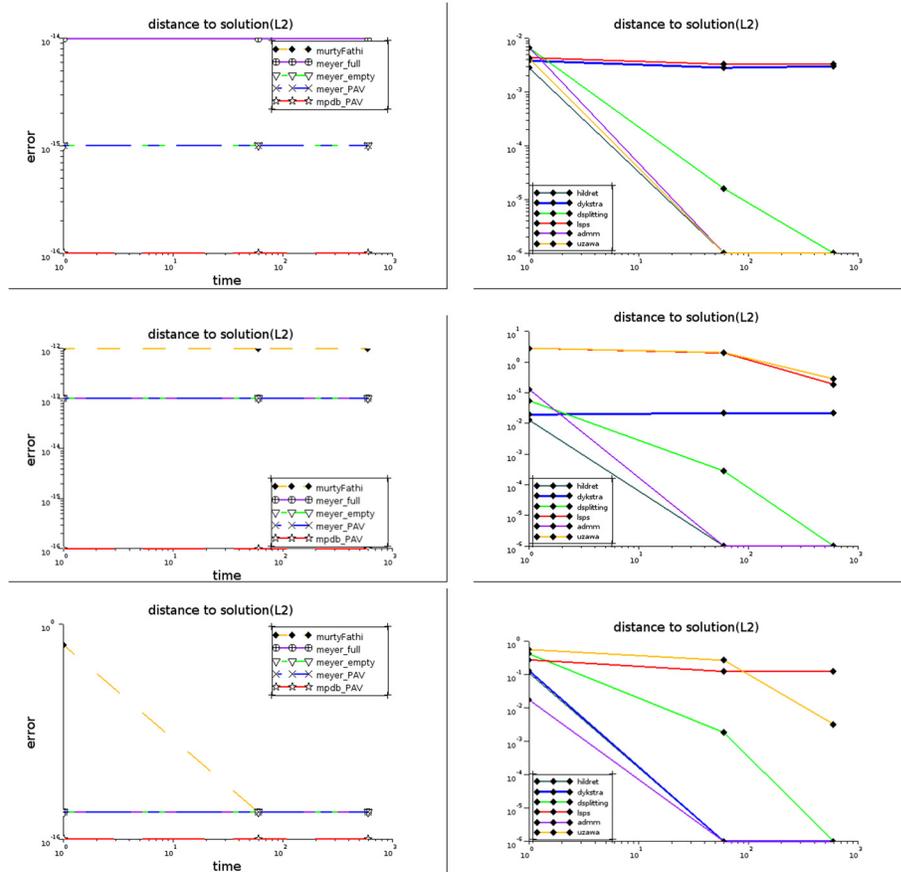


FIG 4. Distance to the solution (L_2 norm) for a signal of type S1 of size 50. From top to the bottom, three increasing level of noise have been added. (Left) Algorithms with finite-time convergence: all them converge instantaneously. (Right) Algorithms with asymptotic convergence: ADMM is more robust to noise. LSPS and Dykstra use a parametrization in the primal parameter space which causes numerical round-off errors to cumulate so that the fitted values do not satisfy the constraints of the dual problem and convergence is not attained.

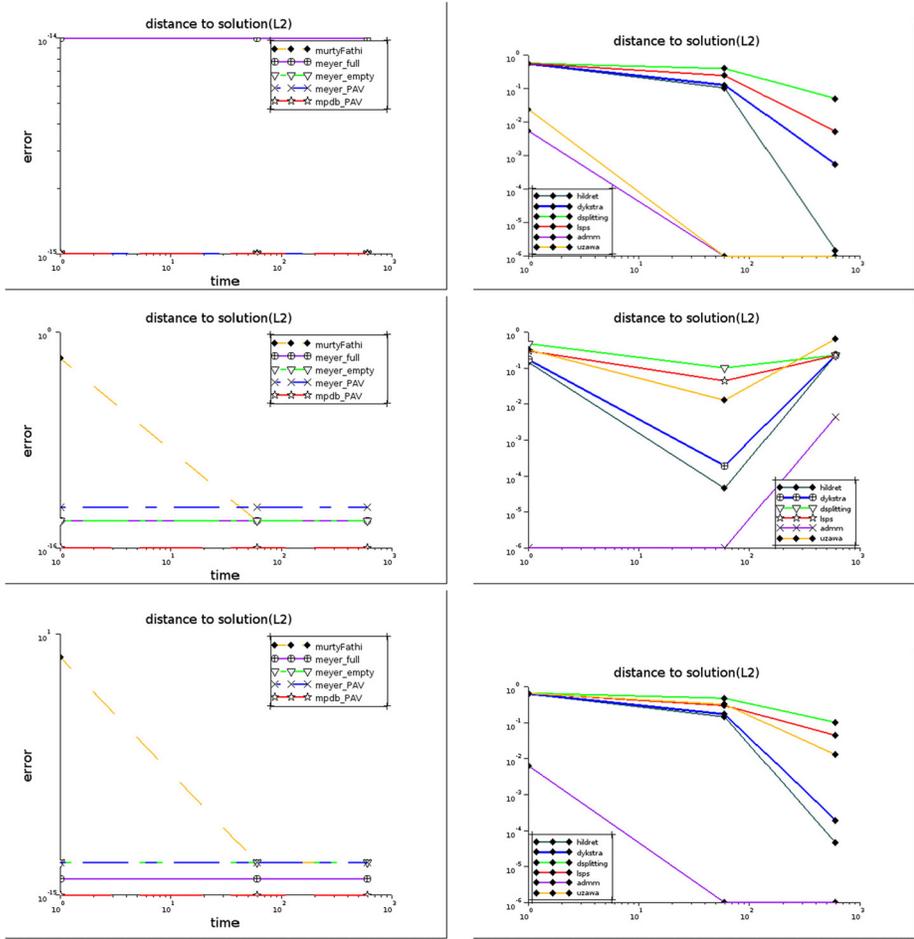


FIG 5. Distance to the solution (L_2 norm) for a signal of type S_2 of size 50. From top to the bottom, three increasing level of noise have been added. (Left) Algorithms with finite-time convergence: all them converge instantaneously. (Right) Algorithms with asymptotic convergence: both ADMM and Hildreth attain the solution when no noise is added but ADMM is much more robust to noise.

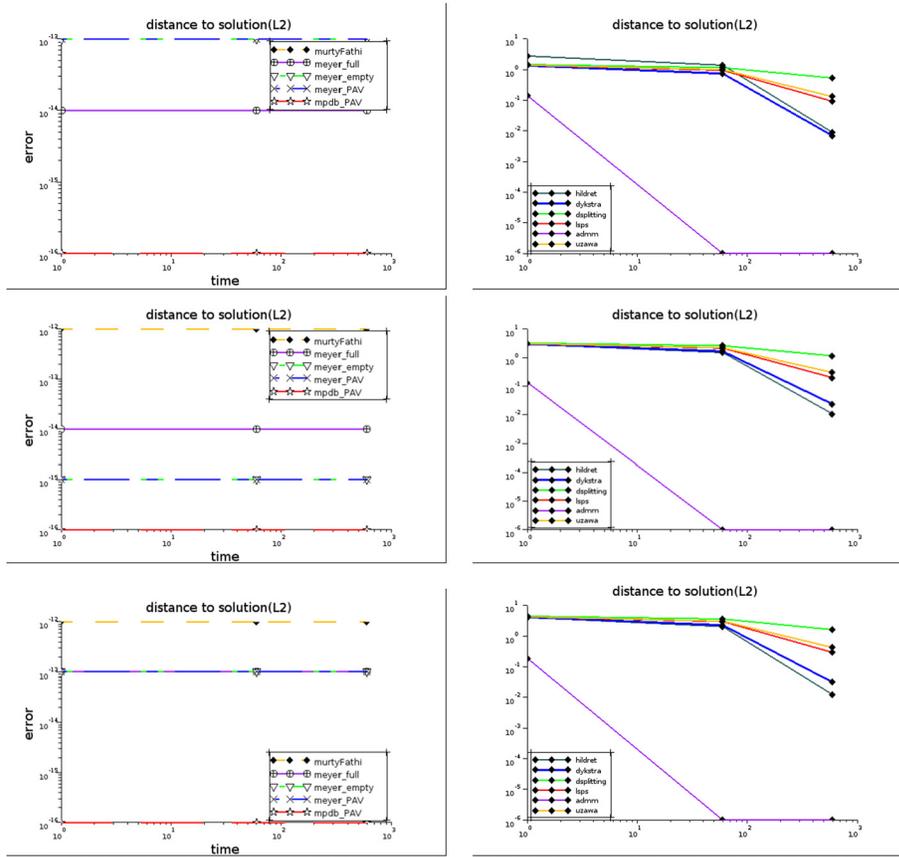


FIG 6. Distance to the solution (L_2 norm) for Gaussian white noise signals of size 50. (Left) Algorithms with finite-time convergence: all them converge instantaneously. (Right) Algorithms with asymptotic convergence: ADMM is the only algorithm that converges.

A.3. Comparative evaluations on signals of size 500

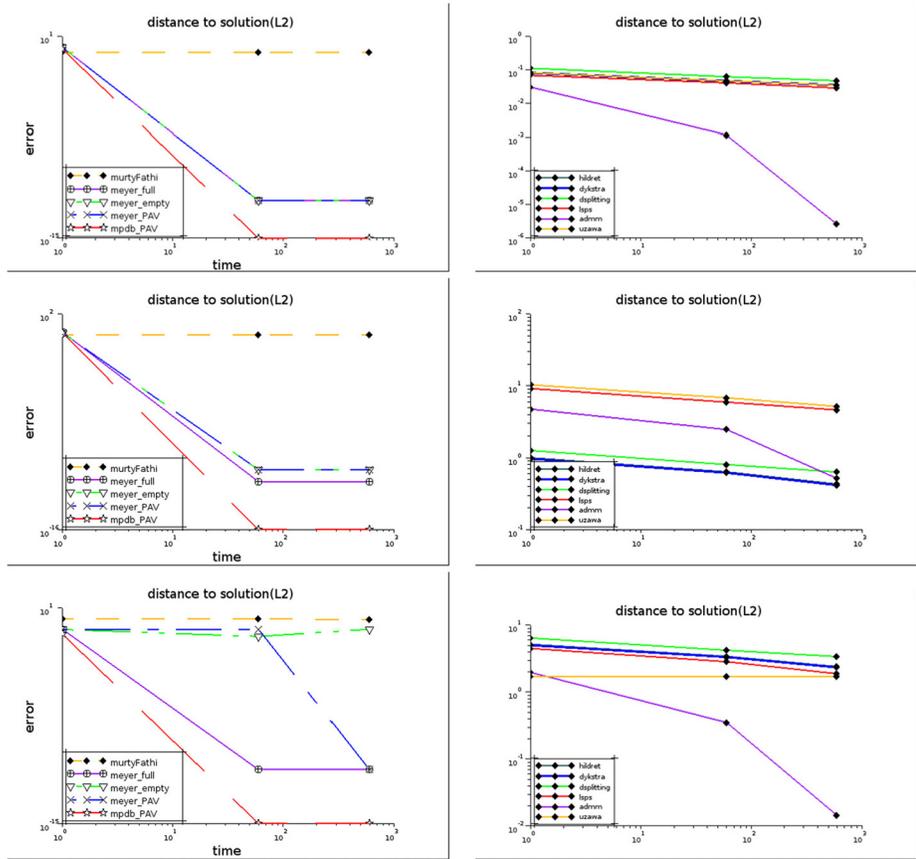


FIG 7. Distance to the solution (L_2 norm) for a signal of type S_1 of size 500. From top to the bottom, three increasing levels of noise have been added. (Left) Algorithms with finite-time convergence: Meyer's algorithm is very sensitive to the initialization. The best performance is achieved by MPDB with a PAV approximate initialization. (Right) Algorithms with asymptotic convergence: ADMM converges only when the signal is slightly noisy.

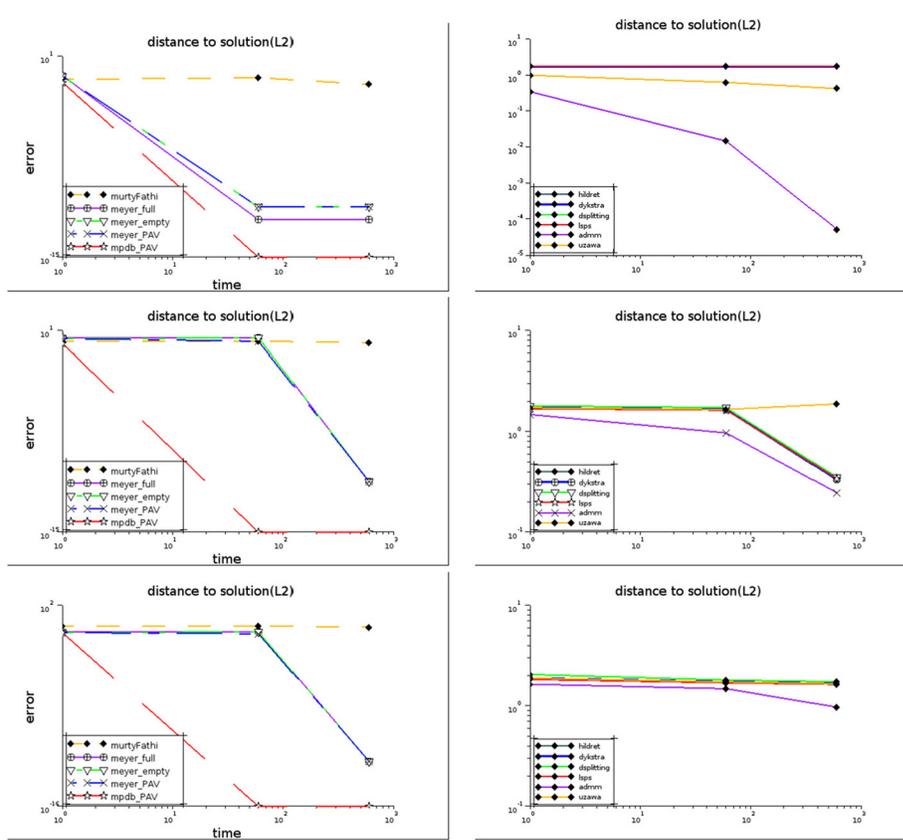


FIG 8. Distance to the solution (L_2 norm) for a signal of type S_2 of size 500. From top to the bottom, three increasing level of noise have been added. (Left) Algorithms with finite-time convergence: the best performance is achieved by MPDB with a PAV's inspired initialization. (Right) Algorithms with asymptotic convergence: no algorithm converges.

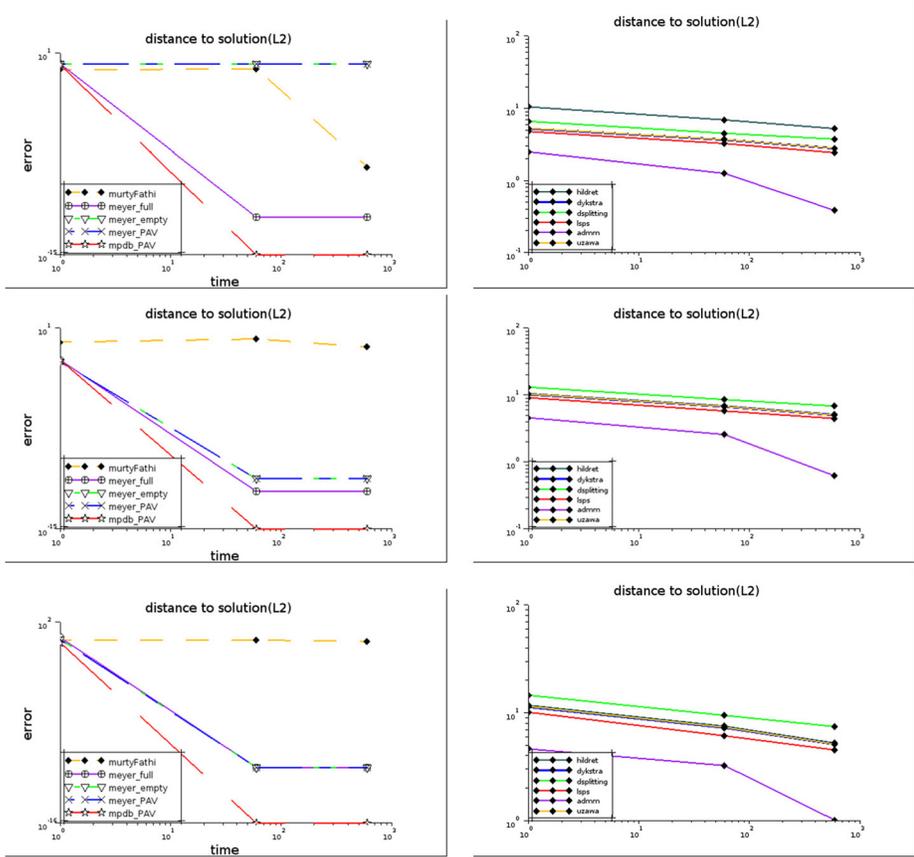


FIG 9. Distance to the solution (L_2 norm) for a Gaussian white noise signal of size 500. (Left) Algorithms with finite-time convergence: the best performance is achieved by MPDB with a PAV's inspired initialization. (Right) Algorithms with asymptotic convergence: no algorithm converges.

A.4. Comparative evaluations on signals of size 1000

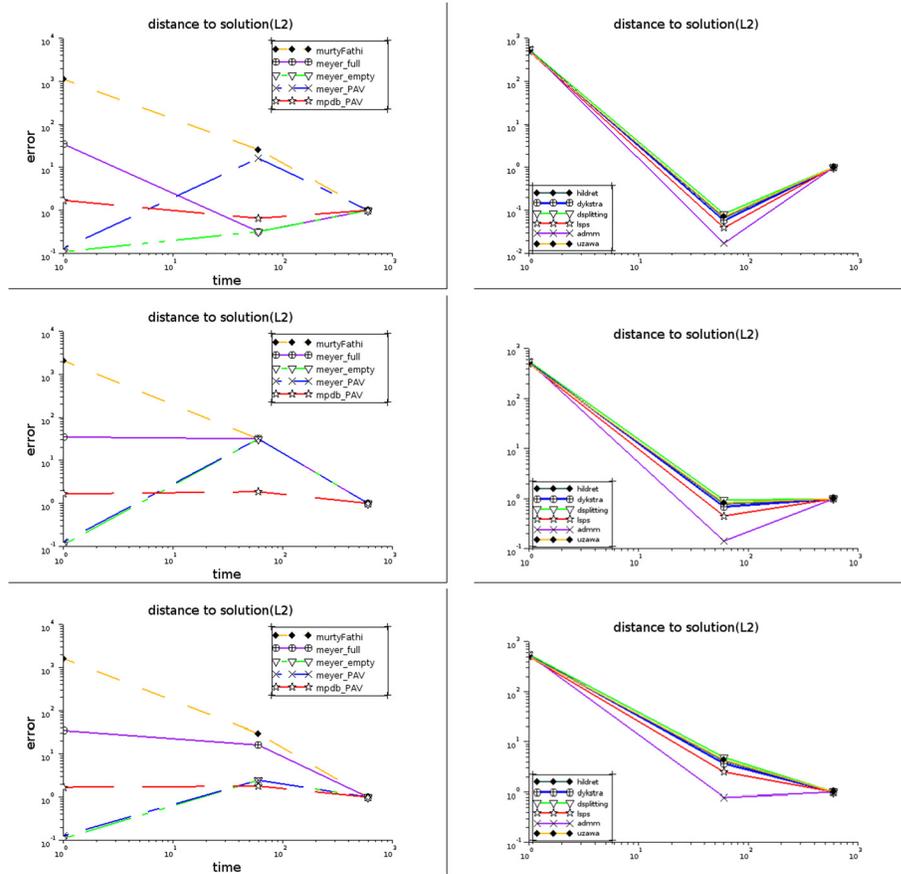


FIG 10. Distance to the solution (L_2 norm) for a signal of type S_1 of size 1000. From top to the bottom, three increasing level of noise have been added. (Left) Algorithms with finite-time convergence: the best performance is achieved by MPDB with a PAV's inspired initialization. (Right) Algorithms with asymptotic convergence: no algorithm converges.

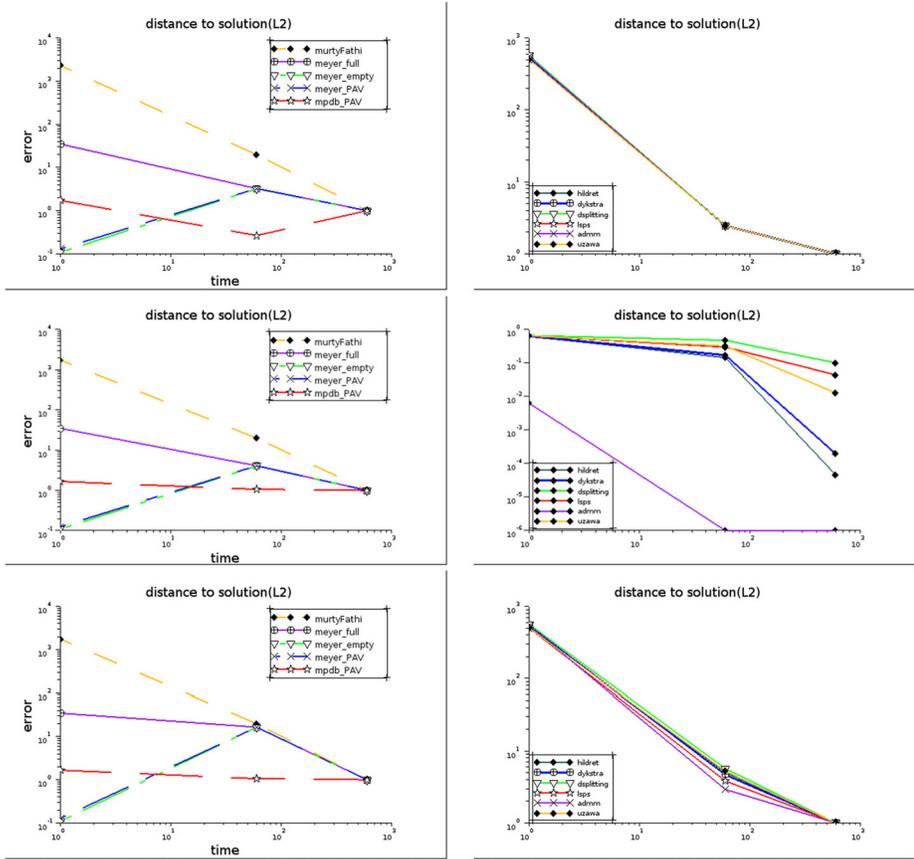


FIG 11. Distance to the solution (L_2 norm) for a signal of type S_2 of size 1000. From top to the bottom, three increasing level of noise have been added. (Left) Algorithms with finite-time convergence: the best performance is achieved by MPDB with a PAV's inspired initialization. (Right) Algorithms with asymptotic convergence: no algorithm converges.

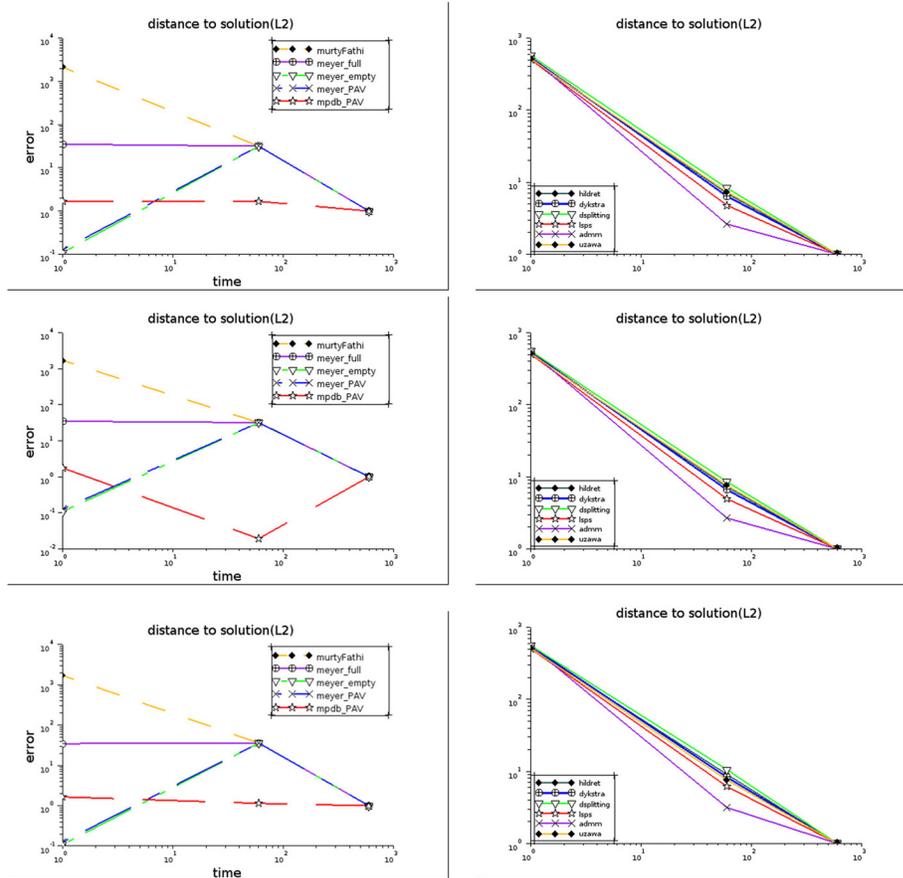


FIG 12. Distance to the solution (L_2 norm) for a signal of type S_3 of size 1000. (Left) Algorithms with finite-time convergence: the best performance is achieved by MPDB with a PAV's inspired initialization when the round-off error does not dominate the calculation and by the Meyer algorithm whose initial active set is empty. This is easy to understand since, for a pure noise signal a high degree of freedom is expected at the solution and therefore the empty active set is close to the final active set. (Right) Algorithms with asymptotic convergence: no algorithm converges.

Appendix B: Pseudocode of the algorithms reviewed

Basic notations:

y is the input signal

N is the iteration number (when required by the algorithm)

\hat{x} is the output signal

Algorithm 1: Hildreth's algorithm

Data: $y \in \mathcal{R}^n$, $A \in \mathcal{R}^{m \times n}$

Result: $\hat{x} = \operatorname{argmin} \|y - x\|^2$ subject to $Ax \leq 0$

begin

$P \leftarrow AA^T$

$b = Ay$

$\lambda_i^0 \leftarrow 0 \quad \forall i = 1, \dots, m$

for $k = 1 : N$ **do**

for $i = 1 : m$ **do**

$\lambda_i^{k+1} = \max(0, (b_i - \sum_{j=1}^{i-1} p_{ij} \lambda_j^k - \sum_{j=i+1}^m p_{ij} \lambda_j^{k-1}) / p_{ii})$

$\hat{x} \leftarrow y - A^T \lambda^N$

end

Algorithm 2: Dykstra (proximal formulation)

Data: $y \in \mathcal{R}^n$, $A \in \mathcal{R}^{m \times n}$, maximum number of iterations N

Result: $\hat{x} = \operatorname{argmin} \|y - x\|^2$ subject to $Ax \leq 0$

begin

$x \leftarrow y$

$I_i \leftarrow 0 \quad \forall i = 1, \dots, m$

for $k = 1 : N$ **do**

for $i = 1 : m$ **do**

$z = x + I_i$

$I_i \leftarrow \operatorname{prox}_{A_i} z$

$x \leftarrow z - I_i$

end

$\operatorname{prox}_{A_i} z = \operatorname{prox}_{\{x \mid A_i x \leq 0\}} z = \frac{A_i^T (z^T A_i^T)}{\|A_i\|^2}$

Algorithm 3: Alternating Directions Method of Multipliers (ADMM)**Data:** $y \in \mathcal{R}^n$, $A \in \mathcal{R}^{m \times n}$, maximum number of iterations N **Result:** $\hat{x} = \operatorname{argmin}_{Ax \leq 0} \|y - x\|^2$, is equivalent to $\hat{x} = \operatorname{argmin}_{Ax=z, z \leq 0} f(x) + g(z)$,

$$f(x) = \|y - x\|^2, g(z) = f \circ A = \frac{1}{2} \|Ax - z\|^2$$

begin

```

 $v^1 \leftarrow 0$ 
 $\lambda^1 \leftarrow 0$ 
 $\tau > 0$ 
for  $k = 1 : N$  do
   $x \leftarrow \operatorname{prox}_f^A(v + \lambda)$ 
   $z \leftarrow Ax$ 
   $v \leftarrow \operatorname{prox}_g(z - \lambda)$ 
   $\lambda \leftarrow \lambda + \tau(v - z)$ 
end

```

end

$$\operatorname{prox}_f^A(v + \lambda) = RA^T(\lambda + \tau v) + Ry, R = (I_n + \tau A^T A)^{-1} \operatorname{prox}_g(z - \lambda) = \min_{z \leq 0} z - \lambda / \tau$$

Algorithm 4: Least Squares in a Product Space (LSPS)**Data:** $y \in \mathcal{R}^n$, $A \in \mathcal{R}^{m \times n}$, maximum number of iterations N , $\lambda > 0$ **Result:** $\hat{x} = \operatorname{argmin} \|y - x\|^2$ subject to $Ax \leq 0$, which is equivalent to requires that $x \in S \cap D$, where D is the diagonal vector subspace, S is a closed convex set corresponding to the cartesian product of the sets (A_i) **begin**

```

 $x^1 \leftarrow y$ 
 $d_i^1 \leftarrow y, i = 1, \dots, m$ 
for  $k = 1 : N$  do
   $x^{k+1} \leftarrow \lambda(\operatorname{prox}_{A_1} d_1^k, \dots, \operatorname{prox}_{A_m} d_m^k)$ 
   $d^{k+1} \leftarrow (\sum_{i=1}^m x_i^{k+1}, \dots, \sum_{i=1}^m x_i^{k+1})$ 
end

```

end

$$\operatorname{prox}_{A_i} d_i = \operatorname{prox}_{\{x | A_i x \leq 0\}} d_i = \frac{A_i^T (d_i^T A_i^T)}{\|A_i\|^2}$$

Algorithm 5: Uzawa**Data:** $y \in \mathcal{R}^n$, $A \in \mathcal{R}^{m \times n}$, maximum number of iterations N **Result:** $\hat{x} = \operatorname{argmin} \|y - x\|^2$ subject to $Ax \leq 0$, which is equivalent to 3.3**begin**

```

 $\rho \geq 0$ 
 $k \leftarrow 1$ 
for  $k = 1 : N$  do
   $x^k = \operatorname{prox}_g y$ 
   $\mu^{k+1} = \max_{\mu \geq 0} \mu^k + \rho Ax^k$ 
end

```

end

$$g(x) = \langle \mu, Ax \rangle$$

$$\operatorname{prox}_g y = \operatorname{argmin}_x \frac{1}{2} \|x - y\|^2 + g(\mu)$$

Algorithm 6: Mixed Primal-Dual Bases (MPDB)**Data:** $(z, y) \in \mathcal{R}^n \times \mathcal{R}^n$, where z is the vector of independent variables, $A \in \mathcal{R}^{m \times n}$.**Result:** $\hat{x} = \operatorname{argmin} \|y - x\|^2$ subject to $Ax \leq 0$ **begin** $\mathbf{1} \in \mathcal{R}^n, \mathbf{1} = (1, \dots, 1)$ $\gamma_{m+1} \leftarrow \mathbf{1} / \|\mathbf{1}\|$ $\bar{z} = \frac{\sum_{i=1}^n m_i z_i}{\sum_{i=1}^n m_i}$ $\gamma_{m+2} \leftarrow (z - \bar{z}\mathbf{1}) / \|z - \bar{z}\mathbf{1}\|$ $V = [\gamma_{m+1}, \gamma_{m+2}]$ $\Gamma = [A_1^T, \dots, A_m^T, V]$ $J \leftarrow \{1, \dots, m\}$ $v \leftarrow VV^\dagger y$ $u \leftarrow y - v$ $\Delta \leftarrow -(\Gamma^T)^{-1}$ $x_0 \leftarrow \sum_{i=1}^m \Delta_i$ $dx \leftarrow Ax_0$ $dy \leftarrow Au$ $stop \leftarrow False$ **while** $stop \neq True$ **do****if** $d_{y_i} \geq 0, i = 1, \dots, m$ **then** $\quad stop \leftarrow True$ **else** $\quad t \leftarrow \min_{i, d_{y_i} < 0} \frac{dx_i}{dx_i - dy_i}$ $\quad x_0 \leftarrow x_0 + t(u - x_0)$ $\quad j \leftarrow \operatorname{argmin}_i \min_{i, d_{y_i} < 0} \frac{dx_i}{dx_i - dy_i}$ **if** $j \in J$ **then** $\quad J \leftarrow J \setminus \{j\}$ **else** $\quad J \leftarrow J \cup \{j\}$ $\quad dx_j \leftarrow 0$ $\quad C = A^T$ $\quad C(J) = \Delta(J)$ $\quad dy \leftarrow C^\dagger u$ $\hat{x} \leftarrow \Delta(J)\Delta(J)^\dagger u + v$ **end** $\Delta(J) = \{\delta_j, j \in J\}$

Algorithm 7: Meyer's Algorithm

Data: $y \in \mathcal{R}^n$, $A \in \mathcal{R}^{m \times n}$, $J \subseteq \{1, \dots, m\}$.

Result: $\hat{x} = \operatorname{argmin} \|y - x\|^2$ subject to $Ax \leq 0$

```

begin
   $stop \leftarrow False$ 
   $b \leftarrow Ay$ 
  if  $j = \operatorname{argmax}_i b_i, b_j > \epsilon$  then
     $J \leftarrow J \cup \{j\}$ 
  else
     $stop \leftarrow True$ 
     $\theta_i \leftarrow 0 \quad \forall i = 1, \dots, m$ 
  while  $stop \neq True$  do
     $a \leftarrow A(J)^\dagger y$ 
    if  $j = \operatorname{argmin}_i a_i, a_j < \epsilon$  then
       $J \leftarrow J \setminus \{j\}$ 
       $stop \leftarrow True$ 
       $\theta_i \leftarrow 0 \quad \forall i = 1, \dots, m$ 
    else
       $\theta \leftarrow A(J)^T a$ 
       $b \leftarrow A(y - \theta)$ 
      if  $j = \operatorname{argmax}_i b_i, b_j > \epsilon, j \notin J$  then
         $J \leftarrow J \cup \{j\}$ 
         $stop \leftarrow False$ 
      else
         $stop \leftarrow True$ 
     $\hat{x} \leftarrow y - \theta$ 
  end
   $A(J) = \{A_j, j \in J\}$ 

```

Algorithm 8: Active index algorithm

Data: $y \in \mathcal{R}^n$, $A \in \mathcal{R}^{m \times n}$, $J \subseteq \{1, \dots, m\}$.
Result: $\hat{x} = \operatorname{argmin} \|y - x\|^2$ subject to $Ax \leq 0$

begin

$\gamma^{m+1} \leftarrow 1/\|1\|$, $\bar{z} \leftarrow \sum_{i=1}^n m_i z_i / \sum_{i=1}^n m_i$, $\gamma^{m+2} \leftarrow (z - \bar{z}1)/\|z - \bar{z}1\|$
 $E \leftarrow [\gamma^{m+1}, \gamma^{m+2}]$, $\Gamma \leftarrow [A_1^T, \dots, A_m^T, E]$
 $p \leftarrow E' E'^{\dagger} y$, $Y \leftarrow y - p$
 $\text{critical} \leftarrow \text{False}$, $\text{solution} \leftarrow \text{False}$, $\text{stop} \leftarrow \text{False}$
 $V_{\text{critical}} \leftarrow \emptyset$, $N = n$

while $\text{stop} == \text{False}$ **do**

if $\text{critical} == \text{True}$ **then**

$V \leftarrow \gamma^k$, $c \leftarrow k$
 $N_{\text{critical}} \leftarrow \operatorname{card}(V_{\text{critical}})$
 $Jk(i) \leftarrow \text{True} \quad \forall i = 1, \dots, N$
 if $V_{\text{critical}} == 0$ **then**
 $Jk(V_{\text{critical}}) \leftarrow \text{False}$
 $J \leftarrow \{i : Jk(i) == \text{True}\}$, $k \leftarrow J(k)$
 $N_{\text{critical}} \leftarrow N_{\text{critical}} + 1$
 $V_{\text{critical}}(N_{\text{critical}}) \leftarrow k$
 $y \leftarrow y - v(y'v)/\|v\|$
 for $i = 1 : m$ **do**
 $\gamma^i \leftarrow \gamma^i - (vv'\gamma^i)/\|v\|$
 $A \leftarrow A \setminus \{A_c\}$
 $\text{critical} \leftarrow \text{False}$

else if $\text{solution} == \text{True}$ **then**
 $x \leftarrow y - x + p$, $\text{stop} \leftarrow \text{True}$

else

for $i = 1 : n$ **do**

if $y^T \gamma_i \leq 0$ **then**
 $V^i = 0$

else if $y^T \gamma_i > 0$ **then**
 $V^i \leftarrow \gamma_i (y^T \gamma_i) / \|\gamma_i\|^2$

$l \leftarrow \operatorname{argmin}_{i \in \{1, \dots, n\}} \{\|V^i - y\|\}$, $\bar{x} \leftarrow V^l$
 $N(\bar{x}) = \{i \in \{1, \dots, n\} : (y - \bar{x})^T \gamma_i > 0\}$
 if $\operatorname{card}(N(\bar{x})) == 0$ **then**
 $\text{stop} \leftarrow \text{True}$

else if $\operatorname{card}(N(\bar{x})) == 1$ **then**
 $i \leftarrow N(\bar{x})$, $\text{critical} \leftarrow \text{True}$

else

$g \in N(\bar{x})$
 $S \leftarrow \{\gamma_l, \gamma_g\}$
 $I(S) \leftarrow \{l, g\}$
 $\hat{y} \leftarrow \Gamma(S) \Gamma(S)^{\dagger} y$
 $\bar{x} \leftarrow \hat{y}$
 $N(\bar{x}) \leftarrow \{i \in \{1, \dots, n\} : (y - \bar{x})^T \gamma_i > 0\}$
 $\text{stop1} \leftarrow \text{False}$
 while $\text{stop1} == \text{False}$ **do**
 $I(S) \leftarrow \{1, \dots, n\} \setminus (S)$
 if $\operatorname{card}(N(\bar{x})) == 0$ **then**
 $\text{stop1} \leftarrow \text{True}$, $\text{stop} \leftarrow \text{True}$

else if $\operatorname{card}(N(\bar{x})) == 1$ **then**
 $i \leftarrow N(\bar{x})$, $\text{stop1} \leftarrow \text{True}$

else if $N(\bar{x}) \cap I(S)$ **then**
 $\hat{y} \leftarrow \Gamma(S) \Gamma(S)^{\dagger} y$
 $S \leftarrow S \cup \{\gamma_g\}$, $I(S) \leftarrow I(S) \cup \{g\}$
 $\bar{x} \leftarrow \hat{y}$

else
 $y(S) \leftarrow \Gamma(S) (\Gamma(S)^T \Gamma(S))^{-1} \Gamma(S)^T y$
 if $y(S) \in \operatorname{pos}(S)$ **then**
 $\bar{x} \leftarrow y(S)$
 $\bar{x} \leftarrow \sum_{j \in I(S)} \alpha_j \gamma_j$
 $y(S) \leftarrow \sum_{j \in I(S)} \gamma_j \gamma_j$
 $\bar{\lambda} \leftarrow \operatorname{argmin}_{j: \gamma_j < 0} \left\{ \frac{\alpha_j}{\alpha_j - \gamma_j} \right\}$
 $\bar{x} \leftarrow (1 - \bar{\lambda}) \bar{x} + \bar{\lambda} y(S)$
 $S \leftarrow S \setminus \{\gamma_k\}$
 $I(S) \leftarrow I(S) \setminus \{k\}$

end

end
 $\Gamma(J) = \{\Gamma_j, j \in J\}$

Acknowledgements

The author would like to thank Lionel Moisan for insightful suggestions. This work has been carried out during a postdoctoral stage in the Laboratory of Applied Mathematics (MAP5, CNRS UMR 8145) at Paris Descartes University.

Supplementary Material

Implementation in Scilab and Matlab of the algorithms reviewed in “Fundamentals of cone regression”

(doi: [10.1214/16-SS114SUPP](https://doi.org/10.1214/16-SS114SUPP); .zip).

References

- [1] C. Hildreth, Point Estimates of Ordinates of Concave Functions, *Journal of the American Statistical Association* 49 (267), 598–619, (1954). [MR0065093](#)
- [2] W. Dent, A Note on Least Squares Fitting of Functions Constrained to be Either non-negative, Nondecreasing or Convex, *Management Science* 20 (1), 130–132, (1973). [MR0337328](#)
- [3] C. A. Holloway, On the Estimation of Convex Functions, *Operations Research* 27 (2), 401–407, (1979).
- [4] C. Perkins, A Convergence Analysis of Dykstra’s Algorithm for Polyhedral Sets, *SIAM Journal on Numerical Analysis* 40 (2), 792–804, (2003). [MR1921678](#)
- [5] N. I. Gould, How good are projection methods for convex feasibility problems?, *Comput. Optim. Appl.* 40 (1), 1–12, (2008). [MR2386586](#)
- [6] Y. Censor, W. Chen, P. L. Combettes, R. Davidi, G. T. Herman, On the Effectiveness of Projection Methods for Convex Feasibility Problems with Linear Inequality Constraints, *Computational Optimization and Applications*, 51 (3), 1065–1088, (2012). [MR2891928](#)
- [7] J. J. Moreau, Decomposition orthogonale d’un espace Hilbertien selon deux cones mutuellement polaires, *Comptes Rendus de l’Académie des Sciences* 255, 238–240, (1962). [MR0139919](#)
- [8] H. W. Kuhn, A. W. Tucker, Nonlinear programming, in: *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, 481–492, (1951). [MR0047303](#)
- [9] S. M. Goldman, P. A. Ruud, Nonparametric Multivariate Regression Subject to Constraint, Preprint. Department of Economics, UCB, (1993).
- [10] J. J. Moreau, Fonctions convexes duales et points proximaux dans un espace Hilbertien, *Comptes Rendus de l’Académie des Sciences (Paris), Série A* 255, 2897–2899, (1962). [MR0144188](#)
- [11] J. J. Moreau, Propriétés des applications *prox*, *Comptes Rendus de l’Académie des Sciences (Paris), Série A* 256, 1069–1071, (1963). [MR0149244](#)

- [12] R. Dykstra, An Algorithm for Restricted Least Squares Regression, *Journal of the American Statistical Association* 78 (384), 837–842, (1983). [MR0727568](#)
- [13] D. R. Wilhelmsen, A Nearest Point Algorithm for Convex Polyhedral Cones and Applications to Positive Linear Approximation, *Mathematics of Computation* 30 (133), (1976). [MR0394439](#)
- [14] B. N. Pshenichny, M. Y. Danilin, *Numerical methods in extremal problems*, Mir, Moscow, (1978). [MR0502886](#)
- [15] C. F. Wu, Some algorithms for concave and isotonic regression, *Studies in the Management Sciences* 19, 105–116, (1982). [MR0723345](#)
- [16] D. A. S. Fraser, H. Massam, A Mixed Primal-Dual Bases Algorithm for Regression under Inequality Constraints. Application to Concave Regression, *Scandinavian Journal of Statistics* 16 (1), 65–74, (1989) . [MR1003969](#)
- [17] M. C. Meyer, An extension of the mixed primal-dual bases algorithm to the case of more constraints than dimensions, *Journal of Statistical Planning and Inference*, 13–31, (1999). [MR1718405](#)
- [18] M. C. Meyer, A Simple New Algorithm for Quadratic Programming with Applications in Statistics, *Communications in Statistics*.
- [19] K. G. Murty, Y. Fathi, A critical index algorithm for nearest point problems on simplicial cones, *Mathematical Programming* 23 (1), 206–215, (1982). [MR0657080](#)
- [20] Z. Liu, Y. Fathi, An active index algorithm for the nearest point problem in a polyhedral cone, *Computational Optimization and Applications* 49 (3), 435–456, (2011). [MR2803860](#)
- [21] W. Kahan, *Gauss-Seidel Methods of Solving Large Systems of Linear Equations*, Ph.D. thesis, University of Toronto (1958). [MR2702985](#)
- [22] J. von Neumann, *Functional Operators, Volume II*, Princeton University Press, USA, (1950).
- [23] J. P. Boyle, R. L. Dykstra, A method for finding projections onto the intersection of convex sets in Hilbert spaces, *Lecture Notes in Statistics* 37, 28–47, (1986). [MR0875647](#)
- [24] H. R. Varian, The Nonparametric Approach to Production Analysis, *Econometrica* 52 (3), 579–597, (1984). [MR0740302](#)
- [25] A. N. Iusem, A. R. Pierro, On the convergence of Han’s method for convex programming with quadratic objective, *Mathematical Programming* 52 (1), 265–284, (1991). [MR1126172](#)
- [26] G. Crombez, Finding projections onto the intersection of convex sets in Hilbert spaces, *Numerical Functional Analysis and Optimization* 16 (5–6), 637–652, (1995). [MR1341103](#)
- [27] N. Gaffke, R. Mathar, A cyclic projection algorithm via duality, *Metrika* 36 (1), 29–54, (1989). [MR0985010](#)
- [28] H. H. Bauschke, J. M. Borwein, A. S. Lewis, On the method of cyclic projections for convex sets in Hilbert space, Preprint, (1994). [MR1442992](#)
- [29] P. L. Combettes, J. C. Pesquet, *Proximal Splitting Methods in Signal Processing*, Fixed-point algorithms for inverse problems in science and engineering, Springer New York, NY, 185–212, (2011). [MR2858838](#)

- [30] S. P. Han, A successive projection method, *Mathematical Programming*, 40 (1), 1–14, (1988). [MR0923692](#)
- [31] P. A. Ruud, Restricted least squares subject to monotonicity and concavity constraints, in: *Advances in Economics and Econometrics: Theory and Applications*, *Econometric Society Monographs*, 3, 166–187, (1997).
- [32] M. Hestenes, Multiplier and gradient methods, *Journal of Optimization Theory and Applications* 4 (5), 303–320, (1969). [MR0271809](#)
- [33] M. J. D. Powell, A method for nonlinear constraints in minimization problems, R. Fletcher Editions, *Optimization*, Academic Press, New York, 283–298, (1969). [MR0272403](#)
- [34] J. Eckstein, D. P. Bertsekas, On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators, *Mathematical Programming*, 55 (3), 293–318, (1992). [MR1168183](#)
- [35] G. Pierra, Decomposition through formalization in a product space, *Mathematical Programming*, 28 (1), 96–115, (1984). [MR0727421](#)
- [36] I. I. Eremin, Fejer mappings and problems of convex optimization, *Sibirskii Matematicheskii Zhurnal*, 10, 1034–1047, (1969). [MR0253750](#)
- [37] K. J. Arrow, L. Hurwicz, H. Uzawa, *Studies in Linear and Non-Linear Programming*, Cambridge University Press, (1958). [MR0108399](#)
- [38] M. J. Silvapulle, P. K. Sen, *Constrained statistical inference: Order, inequality, and shape constraints*, John Wiley & Sons, (2011). [MR2099529](#)
- [39] R. T. Rockafellar, *Convex Analysis (Princeton Landmarks in Mathematics and Physics)*, Princeton University Press, (1970). [MR1451876](#)
- [40] E. H. Zarantonello (Ed.), *Projections on Convex Sets in Hilbert Space and Spectral Theory*, Academic Press, (1971).
- [41] K. G. Murty, *Linear complementarity, linear and nonlinear programming*, Berlin: Heldermann Verlag, (1988). [MR0949214](#)
- [42] T. Robertson, F. Wright, R. L. Dykstra, T. Robertson, Order restricted statistical inference, *Statistical Papers*, 30 (1), 316–316, (1989). [MR0961262](#)
- [43] N. Karmarkar, A new polynomial-time algorithm for linear programming, *Proceedings of the sixteenth annual ACM symposium on Theory of computing*, STOC '84, ACM, New York, NY, USA, 302–311, (1984). [MR0779900](#)
- [44] J. N. Singh, D. Singh, Interior-Point Methods for Linear Programming: A Review, *International Journal of Mathematical Education in Science and Technology*, 33 (3), 405–423, (2002). [MR1912442](#)
- [45] F. Curtis, Z. Han, D. Robinson, A Globally Convergent Primal-Dual Active-Set Framework for Large-Scale Convex Quadratic Optimization, *Computational Optimization and Applications*, 60 (2): 311–341, (2015). [MR3316681](#)
- [46] T. Bhatia, L. T. Biegler, Dynamic Optimization in the Design and Scheduling of Multiproduct Batch Plants, *Industrial and Engineering Chemistry Research*. 35 (7), 2234–2246, (1996).
- [47] E. Andelić, M. Schafföner, M. Katz, S. E. Krüger, A. Wendemuth, Kernel least-squares models using updates of the pseudoinverse, *Neural Computations*, 18 (12), 2928–2935, (2006).

- [48] R. Mahfoudhi, A fast triangular matrix inversion, Proceedings of the World Congress on Engineering, (2012).
- [49] P. Courrieu, Fast computation of Moore-Penrose inverse matrices, Neural Information Processing - Letters and Reviews 8 (2), 25–29, (2008).
- [50] B.-Israel Adi, A Newton-Raphson method for the solution of systems of equations, Journal of Mathematical Analysis and Applications, 15 (2), 243–252, (1966). [MR0205445](#)
- [51] Y. Zhu, X. Li, Recursive least squares with linear constraints, Communications in Information and Systems 7 (3), 287–312, (2007). [MR2516245](#)
- [52] T. Kuosmanen, Representation Theorem for Convex Nonparametric Least Squares, The Econometrics Journal, 11 (2), 308–325, (2008).