A decomposition method for lasso problems with zero-sum constraint

Andrea Cristofari[*]

[*]Department of Civil Engineering and Computer Science Engineering,
University of Rome "Tor Vergata", Via del Politecnico, 1, 00133 Rome, Italy
E-mail: andrea.cristofari@uniroma2.it

Abstract. In this paper, we consider lasso problems with zero-sum constraint, commonly required for the analysis of compositional data in high-dimensional spaces. A novel algorithm is proposed to solve these problems, combining a tailored active-set technique, to identify the zero variables in the optimal solution, with a 2-coordinate descent scheme. At every iteration, the algorithm chooses between two different strategies: the first one requires to compute the whole gradient of the smooth term of the objective function and is more accurate in the active-set estimate, while the second one only uses partial derivatives and is computationally more efficient. Global convergence to optimal solutions is proved and numerical results are provided on synthetic and real datasets, showing the effectiveness of the proposed method. The software is publicly available.

Keywords. Nonlinear programming. Convex programming. Constrained lasso. Decomposition methods. Active-set methods.

1 Introduction

In this paper, we address the following optimization problem:

$$\begin{aligned} &\min \frac{1}{2}\|Ax-y\|^2 + \lambda\|x\|_1 \\ &\sum_{i=1}^n x_i = 0, \end{aligned} \tag{1}$$

where $\|\cdot\|$ denotes the Euclidean norm, $\|\cdot\|_1$ denotes the ℓ_1 norm, $x \in \mathbb{R}^n$ is the variable vector and $A \in \mathbb{R}^{m\times n}$, $y \in \mathbb{R}^m$, $\lambda \in \mathbb{R}$ are given, with $\lambda \ge 0$.

Problem (1) is an extension of the well known lasso problem [43], imposing the *zero-sum constraint* $\sum_{i=1}^n x_i = 0$. This constraint is required in some regression models for *compositional data*, i.e., for data representing percentages, or proportions, of a whole. Applications with data of this type frequently arise in many different fields, such as geology, biology, ecology and economics. For example, in microbiome analysis, datasets are usually normalized and result in compositional data [22, 41].

Let us briefly review the role of the zero-sum constraint in regression analysis for compositional data. Assume we are given a response vector $y \in \mathbb{R}^m$ and an $m \times n$ matrix Z of covariates, where every row of Z is a sample. By definition, compositional data are vectors whose components are non-negative and sum to 1, so we can assume each row of Z belong to the positive simplex $\Delta^+_{n-1} = \{z \in \mathbb{R}^n \colon \sum_{i=1}^n z_i = 1,\, z > 0\}$. Due to the constrained form of the sample space, standard statistical tools designed for unconstrained data cannot be applied to build a regression model. To overcome this issue, *log-contrast models* were proposed in [1, 2], using the following log-ratio transformation of data:

$$W_{ij} = \log\Bigl(\frac{Z_{ij}}{Z_{ir}}\Bigr), \quad i \in \{1,\dots,m\}, \quad j \in \{1,\dots,n\} \setminus \{r\},$$

where the rth component is referred to as *reference component.* Denoting by $W_{\setminus r} \in \mathbb{R}^{m\times(n-1)}$ the matrix with entries W_{ij}, a linear log-contrast model is then obtained as follows:

$$y = W_{\setminus r}\, x_{\setminus r} + \varepsilon,$$

where $x_{\setminus r} = \begin{bmatrix} x_1 & \dots & x_{r-1} & x_{r+1} & \dots & x_n \end{bmatrix}^T \in \mathbb{R}^{n-1}$ is the vector of regression coefficients and $\varepsilon \in \mathbb{R}^m$ is a vector of independent noise with mean zero and finite variances. Using the definition of $W_{\setminus r}$, we can write

$$y_i = \sum_{j\neq r} x_j \log\left(Z_{ij}\right) - \sum_{j\neq r} x_j \log(Z_{ir}) + \varepsilon_i, \quad i = 1,\dots,m.$$

Hence, introducing $x_r = -\sum_{j\neq r} x_j$, we can express the above model in the following symmetric form:

$$y = Ax + \varepsilon, \quad \text{subject to } \sum_{i=1}^n x_i = 0, \tag{2}$$

where $A \in \mathbb{R}^{m\times n}$ is the matrix with $A_{ij} = \log(Z_{ij})$ in position (i,j). Note that the zero-sum constraint has a central role, since it allows the response y to be expressed as a linear combination of log-ratios.

Starting from (2), an ℓ_1-regularized least-square formulation with zero-sum constraint was first considered in [32] for variable selection in high-dimensional spaces, leading to the optimization problem (1). The resulting estimator was shown in [32] to be *model selection consistent* as well as to ensure *scale invariance*, *permutation invariance* and *selection invariance.* Moreover, in [3] it was shown that model (1) is *proportional reference point insensitive*, allowing to overcome some issues related to the choice of a reference point in molecular measurements.

Well established algorithms can be used to solve (1), exploiting the peculiar structure of the problem. In this fashion, an augmented Lagrangian scheme with the subproblem solved by cyclic coordinate descent was proposed in [32], while a coordinate descent strategy based on random selection of variables was proposed in [3]. Moreover, considering more general forms of constrained lasso, an approach based on quadratic programming and an ADMM method were analyzed in [20], a semismooth Newton augmented Lagrangian method was proposed in [17] and path algorithms were designed in [20, 28, 45].

In this paper we propose a decomposition algorithm, named Active-Set Zero-Sum-Lasso (`AS-ZSL`), to efficiently solve problem (1) in a large-scale setting. The first ingredient of our approach is an *active-set technique* to identify the zero variables in the optimal solution, which is expected to be sparse due to the ℓ_1 regularization. The second ingredient is a *2-coordinate descent scheme* to update the variables estimated to be non-zero in the final solution. In more detail, we define two different strategies: the first one uses the whole gradient of the smooth term of the objective function and is more accurate in the identification of the zero variables, while the second one only needs partial derivatives and is computationally more efficient. To balance computational efficiency and accuracy, in the algorithm we use a simple rule to choose between the two strategies, based on the progress in the objective function. The proposed method is proved to converge to optimal solutions and is shown to perform well in the numerical experiments, compared to other approaches from the literature. The `AS-ZSL` software is available at `https://github.com/acristofari/as-zsl`.

The rest of the paper is organized as follows. In Section 2 we introduce the notation and give some optimality results for problem (1), in Section 3 we present the `AS-ZSL` algorithm, in Section 4 we establish global convergence of `AS-ZSL` to optimal solutions, in Section 5 we show numerical results on synthetic and real datasets, in Section 6 we finally draw some conclusions. Moreover, in Appendix A we report some details on the subproblem we need to solve in the algorithm.

2 Preliminaries and optimality results

In this section, we introduce the notation and give some optimality results for problem (1). First note that we did not include an intercept term in model (2), since it can be omitted if the vector y and the columns of A have mean zero.

2.1 Notation

We indicate by e the vector made of all ones (so that the zero-sum constraint $\sum_{i=1}^n x_i = 0$ can be rewritten as $e^T x = 0$). We denote the Euclidean norm, the ℓ_1 norm and the sup norm of a vector v by $\|v\|$, $\|v\|_1$ and $\|v\|_\infty$, respectively. The maximum between a vector v and 0, denoted by $\max\{v, 0\}$, is understood as a vector whose ith entry is $\max\{v_i, 0\}$. Given a matrix M, the element in position (i, j) is indicated with M_{ij}, while M^i is the ith column of M.

Moreover, we define

$$\varphi(x) = \frac{1}{2}\|Ax - y\|^2,$$

so that the objective function of problem (1) can be expressed as

$$f(x) = \varphi(x) + \lambda\|x\|_1.$$

The gradient of φ is denoted by $\nabla\varphi$.

2.2 Optimality conditions

Let us first rewrite (1) as a smooth problem with non-negativity constraints, using a well known variable transformation [20, 39, 43]. More specifically, we can introduce the variables

$x_+, x_- \in \mathbb{R}^n$ in order to split x into positive and negative components, i.e., $x = x_+ - x_-$, with $x_+ = \max\{x, 0\}$ and $x_- = \max\{-x, 0\}$. We obtain the following reformulation:

$$\begin{aligned} &\min \varphi(x_+ - x_-) + \lambda e^T(x_+ + x_-) \\ &e^T(x_+ - x_-) = 0 \\ &x_+ \geq 0 \\ &x_- \geq 0. \end{aligned} \tag{3}$$

Note that the number of variables has doubled, i.e., problem (3) has $2n$ variables. It is easy to obtain the following equivalence between problems (1) and (3).

Lemma 2.1. *Let x^* be an optimal solution of problem* (1). *Then, (x_+^*, x_-^*) is an optimal solution of problem* (3), *where $x_+^* = \max\{x^*, 0\}$ and $x_-^* = \max\{-x^*, 0\}$.*

Vice versa, if (x_+^*, x_-^*) is an optimal solution of problem (3), *then $x^* = x_+^* - x_-^*$ is an optimal solution of problem* (1).

Since problem (3) is convex and all constraints are linear, we can use KKT conditions and state that a point (x_+^*, x_-^*) is an optimal solution of problem (3) if and only if there exist KKT multipliers $\sigma_+^*, \sigma_-^* \in \mathbb{R}^n$ and $\mu^* \in \mathbb{R}$ such that

$$\begin{aligned} &\nabla_{x_+}\varphi(x_+^* - x_-^*) + \lambda e - \mu^* e - \sigma_+^* = 0, \\ &\nabla_{x^-}\varphi(x_+^* - x_-^*) + \lambda e + \mu^* e - \sigma_-^* = 0, \\ &e^T(x_+^* - x_-^*) = 0, \\ &x_+^* \geq 0, \quad x_-^* \geq 0, \\ &\sigma_+^* \geq 0, \quad \sigma_-^* \geq 0, \\ &(\sigma_+^*)^T x_+^* = 0, \quad (\sigma_-^*)^T x_-^* = 0. \end{aligned}$$

From Lemma 2.1 and the fact that $\nabla_{x_+}\varphi(x_+ - x_-) = -\nabla_{x^-}\varphi(x_+ - x_-)$, it follows that a point x^* is an optimal solution of problem (1) if and only if there exist $\sigma_+^*, \sigma_-^* \in \mathbb{R}^n$ and $\mu^* \in \mathbb{R}$ such that

$$\nabla\varphi(x^*) + \lambda e - \mu^* e - \sigma_+^* = 0, \tag{4a}$$

$$\nabla\varphi(x^*) - \lambda e - \mu^* e + \sigma_-^* = 0, \tag{4b}$$

$$e^T x^* = 0, \tag{4c}$$

$$\sigma_+^* \geq 0, \quad \sigma_-^* \geq 0, \tag{4d}$$

$$(\sigma_+^*)_i\, x_i^* = 0, \quad \forall i\colon x_i^* > 0, \tag{4e}$$

$$(\sigma_-^*)_i\, x_i^* = 0, \quad \forall i\colon x_i^* < 0. \tag{4f}$$

The next theorem provides equivalent optimality conditions for problem (1).

Theorem 2.2. *Let x^* be a feasible point of problem* (1). *The following statements are equivalent:*

(a) x^* is an optimal solution of problem (1);

(b) A scalar μ^* (the same as the one appearing in (4)) exists such that

$$\begin{cases} \nabla_i \varphi(x^*) - \mu^* = \lambda, & i \colon x_i^* < 0, \\ \nabla_i \varphi(x^*) - \mu^* = -\lambda, & i \colon x_i^* > 0, \\ |\nabla_i \varphi(x^*) - \mu^*| \le \lambda, & i \colon x_i^* = 0; \end{cases}$$

(c) $\eta^{min}(x^*) \ge \eta^{max}(x^*)$, where

$$\eta^{min}(x) = \min_{i=1,\dots,n} \bigl\{\nabla_i \varphi(x) + \bigl(2\min\{\operatorname{sign}(x_i), 0\} + 1\bigr)\lambda\bigr\},$$
$$\eta^{max}(x) = \max_{i=1,\dots,n} \bigl\{\nabla_i \varphi(x) + \bigl(2\max\{\operatorname{sign}(x_i), 0\} - 1\bigr)\lambda\bigr\}.$$

Proof. We show the following implications.

- **(a) ⇒ (b)**. If x^* is an optimal solution, then (4) holds. If $x_i^* < 0$, from (4d)–(4f) we have $(\sigma_-^*)_i = 0$ and, from (4b), it follows that $\nabla_i \varphi(x^*) - \mu^* = \lambda$. By the same arguments, if $x_i^* > 0$ we have $(\sigma_+^*)_i = 0$ and $\nabla_i \varphi(x^*) - \mu^* = -\lambda$. If $x_i^* = 0$, from (4a), (4b) and (4d) we have $-\lambda \le \nabla_i \varphi(x^*) - \mu^* \le \lambda$.

- **(b) ⇒ (a)**. If (b) holds, then the KKT system (4) is satisfied by setting $\sigma_+^* = \nabla \varphi(x^*) + \lambda e - \mu^* e$ and $\sigma_-^* = -\nabla \varphi(x^*) + \lambda e + \mu^* e$, implying that x^* is an optimal solution.

- **(b) ⇒ (c)**. Let us rewrite $\eta^{\min}(x^*)$ and $\eta^{\max}(x^*)$ as follows:

$$\eta^{\min}(x^*) = \min\Bigl\{\min_{i \colon x_i^* \ge 0} \nabla_i \varphi(x^*) + \lambda, \min_{i \colon x_i^* < 0} \nabla_i \varphi(x^*) - \lambda\Bigr\}, \tag{5a}$$
$$\eta^{\max}(x^*) = \max\Bigl\{\max_{i \colon x_i^* \le 0} \nabla_i \varphi(x^*) - \lambda, \max_{i \colon x_i^* > 0} \nabla_i \varphi(x^*) + \lambda\Bigr\}. \tag{5b}$$

  If (b) holds, we have

$$\nabla_i \varphi(x^*) + \lambda \begin{cases} \ge \nabla_i \varphi(x^*) - \lambda = \mu^*, & i \colon x_i^* < 0, \\ = \mu^*, & i \colon x_i^* > 0, \\ \ge \mu^*, & i \colon x_i^* = 0, \end{cases}$$

  and

$$\nabla_i \varphi(x^*) - \lambda \begin{cases} = \mu^*, & i \colon x_i^* < 0, \\ \le \nabla_i \varphi(x^*) + \lambda = \mu^*, & i \colon x_i^* > 0, \\ \le \mu^*, & i \colon x_i^* = 0. \end{cases}$$

  Therefore, $\eta^{\min}(x^*) \ge \mu^* \ge \eta^{\max}(x^*)$.

- **(c) ⇒ (b)**. Let us consider $\eta^{\min}(x^*)$ and $\eta^{\max}(x^*)$ written as in (5). If (c) holds, let μ^* be any number in $[\eta^{\max}(x^*), \eta^{\min}(x^*)]$. If $\{i \colon x_i^* < 0\} \ne \emptyset$, we can write

$$\begin{aligned} \min_{i \colon x_i^* < 0} \nabla_i \varphi(x^*) - \lambda &\le \max_{i \colon x_i^* < 0} \nabla_i \varphi(x^*) - \lambda \le \max_{i \colon x_i^* \le 0} \nabla_i \varphi(x^*) - \lambda \\ &\le \eta^{\max}(x^*) \le \mu^* \le \eta^{\min}(x^*) \le \min_{i \colon x_i^* < 0} \nabla_i \varphi(x^*) - \lambda. \end{aligned}$$

Therefore, all the inequalities in the above chain are actually equalities, implying that $\mu^* = \min_{i\colon x_i^*<0} \nabla_i\varphi(x^*) - \lambda = \max_{i\colon x_i^*<0} \nabla_i\varphi(x^*) - \lambda$. Namely,

$$\mu^* = \nabla_i\varphi(x^*) - \lambda, \quad \forall i\colon x_i^* < 0,$$

and the first condition of (b) is satisfied. Similarly, if $\{i\colon x_i^* > 0\} \neq \emptyset$, we can write

$$\begin{aligned}\max_{i\colon x_i^*>0} \nabla_i\varphi(x^*) + \lambda \geq \min_{i\colon x_i^*>0} \nabla_i\varphi(x^*) + \lambda \geq \min_{i\colon x_i^*\geq 0} \nabla_i\varphi(x^*) + \lambda \\ \geq \eta^{\min}(x^*) \geq \mu^* \geq \eta^{\max}(x^*) \geq \max_{i\colon x_i^*>0} \nabla_i\varphi(x^*) + \lambda,\end{aligned}$$

implying that

$$\mu^* = \nabla_i\varphi(x^*) + \lambda, \quad \forall i\colon x_i^* > 0,$$

and the second condition of (b) is satisfied. Finally, if $\{i\colon x_i^* = 0\} \neq \emptyset$, we can write

$$\begin{aligned}\min_{i\colon x_i^*=0} \nabla_i\varphi(x^*) + \lambda \geq \min_{i\colon x_i^*\geq 0} \nabla_i\varphi(x^*) + \lambda \geq \eta^{\min}(x^*) \geq \mu^* \geq \eta^{\max}(x^*) \\ \geq \max_{i\colon x_i^*\leq 0} \nabla_i\varphi(x^*) - \lambda \geq \max_{i\colon x_i^*=0} \nabla_i\varphi(x^*) - \lambda,\end{aligned}$$

implying that

$$\nabla_i\varphi(x^*) - \lambda \leq \mu^* \leq \nabla_i\varphi(x^*) + \lambda, \quad \forall i\colon x_i^* = 0,$$

and the third condition of (b) is satisfied.

□

Note that, by point (c) of Theorem 2.2, we can define $\eta^{\max}(x) - \eta^{\min}(x)$ as a measure of optimality violation. Moreover, now we derive an expression for the KKT multiplier μ^* appearing in Theorem 2.2, which will be used in Section 3 to define an active-set estimate.

Theorem 2.3. *If $x^* \neq 0$ is an optimal solution of problem* (1)*, then the corresponding multiplier μ^* appearing in point (b) of Theorem 2.2 is given by*

$$\mu^* = \frac{\displaystyle\sum_{i\colon x_i^*\neq 0} |x_i^*|^p \big(\nabla_i\varphi(x^*) + \lambda \operatorname{sign}(x_i^*)\big)}{\displaystyle\sum_{i\colon x_i^*\neq 0} |x_i^*|^p}, \quad \forall p \geq 0.$$

Proof. Consider the KKT multiplier μ^* appearing in (4), which is the same as the one appearing in point (b) Theorem 2.2. From (4a) and (4b), we obtain

$$\begin{aligned}\sigma_+^* &= \nabla\varphi(x^*) + \lambda e - \mu^* e, \\ \sigma_-^* &= -\nabla\varphi(x^*) + \lambda e + \mu^* e.\end{aligned}$$

Using (4d)–(4f), it follows that μ^* is the unique optimal solution of the following one-dimensional strictly convex problem, for all $p \geq 0$:

$$\min_\mu \frac{1}{2}\Bigg[\sum_{i\colon x_i^*>0} (\nabla_i\varphi(x^*) + \lambda - \mu)^2 (x_i^*)^p + \sum_{i\colon x_i^*<0} (-\nabla_i\varphi(x^*) + \lambda + \mu)^2 (-x_i^*)^p\Bigg].$$

To compute μ^* as the minimizer of the above univariate function, we have to set its derivative to 0, that is,

$$\mu^* = \frac{\displaystyle\sum_{i\colon x^*_i > 0} (x^*_i)^p (\nabla_i \varphi(x^*) + \lambda) + \sum_{i\colon x^*_i < 0} (-x^*_i)^p (\nabla_i \varphi(x^*) - \lambda)}{\displaystyle\sum_{i\colon x^*_i > 0} (x^*_i)^p + \sum_{i\colon x^*_i < 0} (-x^*_i)^p},$$

leading to the expression given in the assertion. □

Let us conclude this section by showing a regularity property of the non-zero feasible points of problem (1), which will be useful in the global convergence analysis of the proposed algorithm.

Proposition 2.4. *Let $\bar x$ be a feasible point of problem* (1) *such that $\bar x_j \ne 0$ for an index $j \in \{1,\dots,n\}$. If there exists $I \subseteq \{1,\dots,n\}$ such that*

$$f(\bar x) = \min_{\xi \in \mathbb{R}} f(\bar x + \xi(e_i - e_j)), \quad \forall i \in I,$$

then there exists $\bar\mu \in \mathbb{R}$ such that

$$\begin{cases} \nabla_i \varphi(\bar x) - \bar\mu = \lambda, & i \in I\colon \bar x_i < 0, \\ \nabla_i \varphi(\bar x) - \bar\mu = -\lambda, & i \in I\colon \bar x_i > 0, \\ |\nabla_i \varphi(\bar x) - \bar\mu| \le \lambda, & i \in I\colon \bar x_i = 0. \end{cases} \tag{6}$$

Proof. For any point $x \in \mathbb{R}^n$ and a (non-zero) direction $d \in \mathbb{R}^n$, let $f'(x; d)$ be the directional derivative of f at x along d. We have

$$\begin{aligned} f'(x; d) = \lim_{\varepsilon \to 0^+} \frac{f(x + \varepsilon d) - f(x)}{\varepsilon} &= \nabla \varphi(x)^T d + \lambda \lim_{\varepsilon \to 0^+} \frac{\|x + \varepsilon d\|_1 - \|x\|_1}{\varepsilon} \\ &= \nabla \varphi(x)^T d + \lambda \Biggl(\sum_{i=1}^n \operatorname{sign}(x_i)\, d_i + \sum_{i\colon x_i = 0} |d_i| \Biggr). \end{aligned} \tag{7}$$

Since $f(\bar x) = \min_{\xi \in \mathbb{R}} f(\bar x + \xi(e_i - e_j))$ for all $i \in I$ by hypothesis, from the convexity of f we get

$$f'(\bar x; e_i - e_j) \ge 0 \quad \text{and} \quad f'(\bar x; e_j - e_i) \ge 0, \quad \forall i \in I. \tag{8}$$

Without loss of generality, we assume that $\bar x_j > 0$ (the proof for the case $\bar x_j < 0$ is identical, except for minor changes). Using (7), we have

$$f'(\bar x; e_i - e_j) = \begin{cases} \nabla_i \varphi(\bar x) - \nabla_j \varphi(\bar x) - 2\lambda, & \text{if } \bar x_i < 0, \\ \nabla_i \varphi(\bar x) - \nabla_j \varphi(\bar x), & \text{if } \bar x_i \ge 0; \end{cases} \tag{9}$$

$$f'(\bar x; e_j - e_i) = \begin{cases} \nabla_j \varphi(\bar x) - \nabla_i \varphi(\bar x) + 2\lambda, & \text{if } \bar x_i \le 0, \\ \nabla_j \varphi(\bar x) - \nabla_i \varphi(\bar x), & \text{if } \bar x_i > 0. \end{cases} \tag{10}$$

Therefore, in view of (8), for all $i \in I$ such that $\bar{x}_i \neq 0$ we can write

$$\nabla_i \varphi(\bar{x}) = \begin{cases} \nabla_j \varphi(\bar{x}) + 2\lambda, & i \in I \colon \bar{x}_i < 0, \\ \nabla_j \varphi(\bar{x}), & i \in I \colon \bar{x}_i > 0. \end{cases}$$

So, we can set $\bar{\mu} = \nabla_j \varphi(\bar{x}) + \lambda$ and the first two conditions of (6) are satisfied. Moreover, with this choice of $\bar{\mu}$ we have $|\nabla_i \varphi(\bar{x}) - \bar{\mu}| = |\nabla_i \varphi(\bar{x}) - \nabla_j \varphi(\bar{x}) - \lambda|$. So, to show that also the last condition of (6) holds, we have to show that

$$0 \leq \nabla_i \varphi(\bar{x}) - \nabla_j \varphi(\bar{x}) \leq 2\lambda, \quad i \in I \colon \bar{x}_i = 0.$$

This follows from (8), (9) and (10). □

Remark 1. *In Proposition 2.4, if $I = \{1, \dots, n\}$, then $\bar{x}$ is an optimal solution of problem* (1)*, according to point (b) of Theorem 2.2.*

3 The algorithm

In this section we propose a decomposition algorithm, named Active-Set Zero-Sum-Lasso (`AS-ZSL`), to efficiently solve problem (1).

The underlying assumptions motivating our approach are that *the optimal solutions are sparse*, due to the sparsity-promoting ℓ_1 regularization, and that *the problem dimension is large*. To face these issues, the proposed method relies on two main ingredients:

(i) an *active-set technique* to estimate the zero variables in the optimal solution,

(ii) a *2-coordinate descent scheme* to update only the variables estimated to be non-zero in the optimal solution.

In the field of constrained optimization, several active-set techniques were proposed to identify the active (or binding) constraints, see, e.g., [4, 6, 12, 13, 15, 18, 19, 25, 26, 38, 40]. Active-set techniques were successfully used also to identify the zero variables in ℓ_1-regularized problems [7, 16, 30, 42, 49, 50] and in ℓ_1-constrained problems [14]. Moreover, *screening rules* were proposed to identify variables that can be discarded in lasso-type problems [21, 44, 51, 52, 48] and *shrinking techniques* have been widely used in some algorithms for machine learning problems to fix subsets of variables [8, 27, 29, 53].

Our approach makes use of the so-called *multiplier functions*, which will be defined later, adapting the active-set technique originally proposed in [18] for non-linearly constrained problems.

To explain our active-set technique, let us first define the following index sets for any optimal solution x^* of problem (1):

$$\begin{aligned} \bar{\mathcal{A}}(x^*) &= \{i \colon x^*_i = 0\}, \\ \bar{\mathcal{N}}(x^*) &= \{1, \dots, n\} \setminus \bar{\mathcal{A}}(x^*) = \{i \colon x^*_i \neq 0\}. \end{aligned}$$

We say that $\bar{\mathcal{A}}(x^*)$ and $\bar{\mathcal{N}}(x^*)$ represent the *active set* and the *non-active set*, respectively, at x^*. In any point x^k produced by the algorithm, we get estimates of $\bar{\mathcal{A}}(x^*)$ and $\bar{\mathcal{N}}(x^*)$

exploiting point (b) of Theorem 2.2. Namely, we set the vector π^k as an approximation of $\nabla\varphi(x^k) - \mu^* e$ and define

$$\mathcal{A}^k = \{i \colon x_i^k = 0,\, |\pi_i^k| \le \lambda\}, \tag{11a}$$

$$\mathcal{N}^k = \{1, \dots, n\} \setminus \mathcal{A}^k, \tag{11b}$$

as estimates of $\bar{\mathcal{A}}(x^*)$ and $\bar{\mathcal{N}}(x^*)$, respectively.

Once $\mathcal{A}^k$ and $\mathcal{N}^k$ have been computed, we want to move only the variables estimated to be non-zero in the final solution, i.e., the variables in $\mathcal{N}^k$, thus working in a lower dimensional space. Moreover, to efficiently address large-scale problems, a 2-coordinate descent scheme is used for the variable update, that is, we move two coordinates at a time (this is the minimum number of variables we can move to maintain feasibility, due to the equality constraint).

Given a feasible point $x^k \ne 0$ produced by the algorithm, in the sequel we define two possible strategies to compute π^k in (11) and to update the variables. The first strategy uses the whole gradient $\nabla\varphi$ and is more accurate in the active-set estimate, while the second strategy only uses partial derivatives and is computationally more efficient. The rationale is trying to balance accuracy and computational efficiency, in order to calculate the whole gradient vector $\nabla\varphi$ only when needed. In particular, in our algorithm we never compute the matrix $A^T A$, since this is impractical for large dimensions. So, if the residual is known, $\mathcal{O}(m)$ operations are needed to compute a single partial derivative, while $\mathcal{O}(mn)$ operations are needed to compute $\nabla\varphi$.

3.1 Strategy MVP

In this first strategy, we need to compute the whole gradient $\nabla\varphi(x^k)$. Then, to obtain π^k in (11), we use an approximation of the KKT multiplier μ^* by means of the so called multiplier functions. More precisely, given a neighborhood X of an optimal solution x^*, we say that $\mu \colon X \to \mathbb{R}$ is a multiplier function if it is continuous in x^* and such that $\mu(x^*) = \mu^*$. A class of multiplier functions can be straightforwardly obtained from Theorem 2.3. Namely, for all $x \ne 0$, we can define

$$\mu(x) = \frac{\displaystyle\sum_{i=1}^n |x_i|^p \bigl(\nabla_i\varphi(x) + \lambda \operatorname{sign}(x_i)\bigr)}{\displaystyle\sum_{i=1}^n |x_i|^p}, \quad p > 0. \tag{12}$$

Once $\nabla\varphi(x^k)$ and $\mu(x^k)$ have been computed, we can set

$$\pi^k = \nabla\varphi(x^k) - \mu(x^k)e. \tag{13}$$

With this choice of π^k we have the following identification property, ensuring that, if we are sufficiently close to an optimal solution x^*, then $i \in \mathcal{A}^k \Rightarrow x_i^* = 0$, while the inverse implication (i.e., $x_i^* = 0 \Rightarrow i \in \mathcal{A}^k$) holds if $|\nabla_i\varphi(x^*) - \mu^*| < \lambda$.

Proposition 3.1. *Let $\mathcal{A}^k$ and $\mathcal{N}^k$ be defined as in* (11)*, with π^k computed as in* (13) *and $\mu(x^k)$ computed as in* (12)*. Then, for any optimal solution x^* of problem* (1)*, there exists a neighborhood $\mathcal{B}(x^*)$ such that*

$$\bar{\mathcal{A}}^+(x^*) \subseteq \mathcal{A}^k \subseteq \bar{\mathcal{A}}(x^*), \quad \forall x^k \in \mathcal{B}(x^*),$$

where $\bar{\mathcal{A}}^+(x^*) = \{i \colon x^*_i = 0, |\nabla_i \varphi(x^*) - \mu^*| < \lambda\}$ and μ^* is the KKT multiplier.

Proof. The inclusion $\mathcal{A}^k \subseteq \bar{\mathcal{A}}(x^*)$ is trivial, since $x^*_i \neq 0$ implies that, for all x^k in a neighborhood of x^*, we have $x^k_i \neq 0$ and then $i \notin \mathcal{A}^k$. The inclusion $\bar{\mathcal{A}}^+(x^*) \subseteq \mathcal{A}^k$ follows from the continuity of $\nabla \varphi(x)$ and the continuity of $\mu(x)$, recalling that $\mu(x^*) = \mu^*$. □

To update the variables in $\mathcal{N}^k$, we use a 2-coordinate descent scheme based on the so called *maximal violating pair* (MVP), i.e., we move the two variables that most violate an optimality measure. In the literature, similar choices were considered for singly linearly constrained problems when the objective function is smooth, using proper optimality measures (see, e.g., [5, 29, 31, 35, 36]). In our case, using point (c) of Theorem 2.2, a maximal violating pair in $\mathcal{N}^k$ is defined as any pair $(\hat{\imath}, \hat{\jmath})$ such that

$$\hat{\imath} \in \operatorname*{Argmin}_{i \in \mathcal{N}^k} \bigl\{\nabla_i \varphi(x^k) + \bigl(2 \min\{\operatorname{sign}(x^k_i), 0\} + 1\bigr)\lambda\bigr\}, \tag{14a}$$

$$\hat{\jmath} \in \operatorname*{Argmax}_{j \in \mathcal{N}^k} \bigl\{\nabla_j \varphi(x^k) + \bigl(2 \max\{\operatorname{sign}(x^k_j), 0\} - 1\bigr)\lambda\bigr\}. \tag{14b}$$

The next feasible point x^{k+1} is then obtained by minimizing f with respect to $x_{\hat{\imath}}$ and $x_{\hat{\jmath}}$, keeping all the other variables fixed in x^k, that is,

$$x^{k+1} \in \operatorname{Argmin}\{f(x) \colon e^T x = 0, \, x_h = x^k_h, h \in \{1, \dots, n\} \setminus \{\hat{\imath}, \hat{\jmath}\}\}.$$

Equivalently,

$$\begin{aligned} & x^{k+1} = x^k + \xi^* (e_{\hat{\imath}} - e_{\hat{\jmath}}), \\ & \xi^* \in \operatorname*{Argmin}_{\xi \in \mathbb{R}} \{f(x^k + \xi(e_{\hat{\imath}} - e_{\hat{\jmath}}))\}. \end{aligned} \tag{15}$$

3.2 Strategy AC2CD

This second strategy does not require to compute the whole gradient $\nabla \varphi$. In (11) we simply set $\pi^k = \pi^{k-1}$ and then we update the variables in $\mathcal{N}^k$ by means of an *almost cyclic rule.* In particular, we extend a 2-coordinate descent method, named AC2CD, proposed in [10] and further analyzed in [11]. Considering singly linearly constrained problems with lower and upper bound on the variables, the method proposed in [10] chooses two variables at a time, such that one of them must be “sufficiently far” from the lower and the upper bound in some points produced by the algorithm, while the other one is picked cyclically. In order to adapt this approach to our setting, we first have to choose a variable index $j(k)$ such that

$$|x^k_{j(k)}| \geq \tau \|x^k\|_\infty, \quad j(k) \in \mathcal{N}^k, \tag{16}$$

where $\tau \in (0,1]$ is a fixed parameter. Namely, we require $x^k_{j(k)}$ to be "sufficiently different" from zero. Then, we start a cycle of inner iterations where we update two variables at a time. In particular, first we set $z^{k,1} = x^k$ and choose a permutation $p^k_1, \dots, p^k_{|\mathcal{N}^k|}$ of $\mathcal{N}^k$, then we select one index p^k_i at a time in a cyclic fashion, with $i = 1, \dots, |\mathcal{N}^k|$, and consider the index pair $(p^k_i, j(k))$. We compute $z^{k,i+1}$ by minimizing $f(z)$ with respect to $z_{p^k_i}$ and $z_{j(k)}$, keeping all the other variables fixed in $z^{k,i}$. Namely,

$$z^{k,i+1} \in \operatorname{Argmin}\{f(z) \colon e^T z = 0,\, z_h = z^{k,i}_h, h \in \{1,\dots,n\} \setminus \{p^k_i, j(k)\}\}.$$

Equivalently,

$$\begin{aligned} & z^{k,i+1} = z^{k,i} + \xi^*(e_{p^k_i} - e_{j(k)}), \\ & \xi^* \in \operatorname*{Argmin}_{\xi \in \mathbb{R}}\{f(z^{k,i} + \xi(e_{p^k_i} - e_{j(k)}))\}. \end{aligned} \tag{17}$$

After producing the points $z^{k,1}$, $z^{k,2}$, $\dots$, $z^{k,|\mathcal{N}^k|+1}$, we set the next feasible point $x^{k+1} = z^{k,|\mathcal{N}^k|+1}$.

3.3 Choosing between the two strategies

We have seen that, one the one hand, Strategy AC2CD does not need the expensive computation of the whole gradient $\nabla\varphi$, but, on the other hand, we expect the active-set estimate used in Strategy MVP to be more accurate in a neighborhood of an optimal solution, according to Proposition 3.1. In order to balance computational efficiency and accuracy, we want to use Strategy MVP only when we judge it is worthwhile to compute a new gradient and a new π^k in (11). This occurs when we observe no sufficient progress in the objective function. In particular, given a parameter $\theta \in (0,1]$, at iteration k we use Strategy MVP if both $f(x^{k-1}) - f(x^k) \le \theta \max\{f(x^{k-1}), 1\}$ and Strategy MVP was not used in x^{k-1}, otherwise we use Strategy AC2CD. As to be shown below, eventually the algorithm alternates between the two strategies.

The scheme of the proposed `AS-ZSL` method is reported in Algorithm 1.

Remark 2. *Both Strategy MVP and Strategy AC2CD require to solve a subproblem for every variable update, given in* (15) *and* (17)*, respectively. We see that each subproblem consists in an exact minimization of f over a direction of the form $\pm(e_i - e_j)$. A finite procedure for this minimization is described in Appendix A.*

4 Convergence analysis

In this section, we show the convergence of `AS-ZSL` to optimal solutions, using some results on the proposed active-set estimate and standard arguments on block coordinate descent methods [23, 33, 46].

First, we note that most results stated in the above sections require $x^k \neq 0$. Indeed, `AS-ZSL` ensures that

$$x^k \neq 0, \quad \forall k \ge 1,$$

Algorithm 1 Active-Set Zero-Sum-Lasso (`AS-ZSL`)

```
 0 Given $\theta \in (0,1]$ and $\tau \in (0,1]$, set $x^0 = 0$ and $k = 0$
 1 If $x^0$ is not optimal, compute $x^1$ such that $f(x^1) < f(x^0)$, set $k = 1$ and go to line 4

 2 While $x^k$ is not optimal

 3     If $\dfrac{f(x^{k-1}) - f(x^k)}{\max\{f(x^{k-1}), 1\}} \le \theta$ and Strategy MVP was not used for $x^{k-1}$

           Strategy MVP
 4         Compute $\mathcal{A}^k$ and $\mathcal{N}^k$ as in (11), with $\pi^k$ computed as in (13)
 5         Compute a maximal violating pair $(\hat{\imath}, \hat{\jmath})$ as in (14)
 6         Compute $x^{k+1}$ as in (15)

 7     Else

           Strategy AC2CD
 8         Compute $\mathcal{A}^k$ and $\mathcal{N}^k$ as in (11), with $\pi^k = \pi^{k-1}$
 9         Choose a variable index $j(k)$ satisfying (16)
10         Set $z^{k,1} = x^k$
11         Choose a permutation $\{p_1^k, \dots, p_{|\mathcal{N}^k|}^k\}$ of $\mathcal{N}^k$
12         For $i = 1, \dots, |\mathcal{N}^k|$
13             Compute $z^{k,i+1}$ as in (17)
14         End for
15         Set $x^{k+1} = z^{k,|\mathcal{N}^k|+1}$

16     End if

17     Set $k = k + 1$
18 End while
```

since $x^0 = 0$ and

$$f(x^{k+1}) \le f(x^k) < f(x^0) = f(0), \quad \forall k \ge 1. \tag{18}$$

As a consequence, for all $k \ge 1$,

- the multiplier function $\mu(x)$ given in (12) is well defined at x^k;
- $\mathcal{N}^k \neq \emptyset$, according to (11).

Since f is continuous and coercive, also the following result follows from (18), ensuring the convergence of $\{f(x^k)\}$ and the existence of limit points for $\{x^k\}$.

Lemma 4.1. *Let $\{x^k\}$ be an infinite sequence of points produced by* `AS-ZSL`*. Then,*

(i) $\lim_{k\to\infty} f(x^k) = f^* \in \mathbb{R}$*, with* $f^* > 0$*;*

(ii) every subsequence $\{x^k\}_{K\subseteq\mathbb{N}}$ has limit points, each of them being feasible and different from zero.

Moreover, for any iteration k where Strategy AC2CD is used, from the instructions of the algorithm we have

$$f(x^{k+1}) = f(z^{k,|\mathcal{N}^k|+1}) \le f(z^{k,|\mathcal{N}^k|}) \le \ldots \le f(z^{k,1}) = f(x^k).$$

So, still using (18) and the fact that f is continuous and coercive, we have the following result. It ensures, over a subsequence of points produced by Strategy AC2CD, the existence of limit points and the convergence of the objective function.

Lemma 4.2. *Let $\{x^k\}_{K\subseteq\mathbb{N}}$ be an infinite subsequence of points produced by* **`AS-ZSL`** *such that Strategy AC2CD is used for all $k \in K$. Then, for every fixed $i \in \{1,\ldots,\min\limits_{k\in K} |\mathcal{N}^k| + 1\}$,*

(i) $\lim\limits_{\substack{k\to\infty \\ k\in K}} f(z^{k,i}) = \lim\limits_{k\to\infty} f(x^k) = f^* \in \mathbb{R}$, with $f^* > 0$;

(ii) the subsequence $\{z^{k,i}\}_K$ has limit points, each of them being feasible and different from zero.

Now, we show that `AS-ZSL` cannot select Strategy MVP or Strategy AC2CD for an arbitrarily large number of consecutive iterations. In particular, provided $\{x^k\}$ is infinite, eventually the algorithm alternates between the two strategies.

Proposition 4.3. *Let $\{x^k\}$ be an infinite sequence of points produced by* **`AS-ZSL`***. Then, there exists $\bar{k}$ such that Strategy MVP is used for $k = \bar{k}, \bar{k}+2, \bar{k}+4, \ldots$ and Strategy AC2CD is used for $k = \bar{k}+1, \bar{k}+3, \bar{k}+5, \ldots$.*

Proof. From point (i) of Lemma 4.1 and (18), the sequence $\{f(x^k)\}$ converges to a value f^* such that $f(x^k) \ge f^*$ for all $k \ge 0$. It follows that

$$\lim_{k\to\infty} \frac{f(x^{k-1}) - f(x^k)}{\theta \max\{f(x^{k-1}), 1\}} \le \lim_{k\to\infty} \frac{f(x^{k-1}) - f(x^k)}{\theta \max\{f^*, 1\}} = 0.$$

Therefore, according to the test at line 3 of Algorithm 1, the algorithm alternates between Strategy MVP and Strategy AC2CD infinitely for sufficiently large k. □

4.1 Global convergence to optimal solutions

Without loss of generality, for every index pair (i,j) selected by Strategy MVP or Strategy AC2CD, now we require f to be strictly convex along the directions $\pm(e_i - e_j)$. In Appendix A.1, we show how this can be easily guaranteed. Essentially, we just have to remove variables from the problem when we find identical columns in the matrix A. Using this non-restrictive requirement, the following results show that $\|x^{k+1} - x^k\| \to 0$.

Proposition 4.4. *Let $\{x^k\}_{K_1 \subset \mathbb{N}}$ and $\{x^k\}_{K_2 \subset \mathbb{N}}$ be two infinite subsequences of points produced by* `AS-ZSL`*, such that Strategy MVP is used for all $k \in K_1$ and Strategy AC2CD is used for all $k \in K_2$. Then,*

$$\lim_{\substack{k\to\infty \\ k\in K_1}} \|x^{k+1} - x^k\| = 0, \tag{19a}$$

$$\lim_{\substack{k\to\infty \\ k\in K_2}} \|x^{k+1} - x^k\| = \lim_{\substack{k\to\infty \\ k\in K_2}} \sum_{i=1}^{|\mathcal{N}^k|} \|z^{k,i+1} - z^{k,i}\| = 0. \tag{19b}$$

Proof. By contradiction, assume that (19a) does not hold. It follows that there exists a subsequence $\{x^k\}_{K_3 \subseteq K_1}$ such that $\liminf_{k\to\infty,\, k\in K_3} \|x^{k+1} - x^k\| > 0$. By point (ii) of Lemma 4.1, we can assume that both $\{x^k\}_{K_3}$ and $\{x^{k+1}\}_{K_3}$ converge to feasible points (passing into a further subsequence if necessary), so that

$$\lim_{\substack{k\to\infty \\ k\in K_3}} x^k = x' \neq x'' = \lim_{\substack{k\to\infty \\ k\in K_3}} x^{k+1}. \tag{20}$$

Since the set of variable indices is finite, we can also assume that the maximal violating pair $(\hat\imath, \hat\jmath)$ is the same for all $k \in K_3$ (passing again into a further subsequence if necessary). Using (15), (20) and the continuity of f, we obtain

$$x'' = x' + \xi^*(e_{\hat\imath} - e_{\hat\jmath}), \quad \text{where} \quad \xi^* \in \operatorname*{Argmin}_{\xi\in\mathbb{R}}\{f(x' + \xi(e_{\hat\imath} - e_{\hat\jmath}))\}.$$

Namely, x' and x'' belong to the line $\{x' + \xi(e_{\hat\imath} - e_{\hat\jmath}),\, \xi \in \mathbb{R}\}$. As said at the beginning of this subsection, without loss of generality here we require f to be strictly convex along $\pm(e_{\hat\imath} - e_{\hat\jmath})$ (in Appendix A.1 it is shown how this can be easily guaranteed). Since $x' \neq x''$, it follows that

$$f\Big(\frac{x' + x''}{2}\Big) < \frac{1}{2} f(x') + \frac{1}{2} f(x''). \tag{21}$$

Moreover, using again (15) we can write

$$f(x^{k+1}) \le f\Big(x^k + \frac{1}{2}(x^{k+1} - x^k)\Big) = f\Big(\frac{x^k + x^{k+1}}{2}\Big) \le \frac{f(x^k)}{2} + \frac{f(x^{k+1})}{2},$$

where the last inequality follows from the convexity of f. Since $f(x^{k+1}) \le f(x^k)$, we obtain

$$f(x^{k+1}) \le f\Big(\frac{x^k + x^{k+1}}{2}\Big) \le f(x^k). \tag{22}$$

By (20) and point (i) of Lemma 4.1, using the continuity of f we have that

$$f(x') = \lim_{\substack{k\to\infty \\ k\in K_3}} f(x^k) = \lim_{\substack{k\to\infty \\ k\in K_3}} f(x^{k+1}) = f(x'').$$

Therefore, taking the limits in (22), we get $f\Big(\dfrac{x' + x''}{2}\Big) = f(x') = f(x'')$, that is,

$$f\Big(\frac{x' + x''}{2}\Big) = \frac{1}{2} f(x') + \frac{1}{2} f(x''),$$

contradicting (21).

To show (19b), from the instructions of the algorithm we can write $x^{k+1}-x^k = \sum_{i=1}^{|\mathcal{N}^k|}(z^{k,i+1}-z^{k,i})$ for all $k \in K_2$, implying that

$$\|x^{k+1}-x^k\| \le \sum_{i=1}^{|\mathcal{N}^k|}\|z^{k,i+1}-z^{k,i}\|, \quad \forall k \in K_2.$$

So, to prove the desired result, we just have to show that

$$\lim_{\substack{k\to\infty\\ k\in K_2}} \sum_{i=1}^{|\mathcal{N}^k|}\|z^{k,i+1}-z^{k,i}\| = 0.$$

Arguing by contradiction, assume that there exist an index $\bar\imath \in \{1,\dots,n\}$ and a subsequence $\{z^{k,\bar\imath}\}_{K_4\subseteq K_2}$ such that $\liminf_{k\to\infty,\, k\in K_4}\|z^{k,\bar\imath+1}-z^{k,\bar\imath}\| > 0$ for all $k \in K_4$. By point (ii) of Lemma 4.2, we can assume that both $\{z^{k,\bar\imath}\}_K$ and $\{z^{k,\bar\imath+1}\}_K$ converge to feasible points (passing into a further subsequence if necessary). Thus, we get a contradiction by the same arguments used to prove (19a). □

Proposition 4.5. *Let $\{x^k\}$ be an infinite sequence of points produced by* `AS-ZSL`*. Then,*

$$\lim_{k\to\infty}\|x^{k+1}-x^k\| = 0.$$

Proof. From Proposition 4.3, two infinite subsequences $\{x^k\}_{K_1\subset\mathbb{N}}$ and $\{x^k\}_{K_2\subset\mathbb{N}}$ exist such that Strategy MVP is used for all $k \in K_1$ and Strategy AC2CD is used for all $k \in K_2$, with $K_1 \cup K_2 = \mathbb{N}$. Then, the desired result follows from Proposition 4.4. □

We are finally ready to show global convergence of `AS-ZSL` to optimal solutions.

Theorem 4.6. *Let $\{x^k\}$ be an infinite sequence of points produced by* `AS-ZSL`*. Then, $\{x^k\}$ has limit points and every limit point is an optimal solution of problem* (1)*.*

Proof. From point (ii) of Lemma 4.1, $\{x^k\}$ has limit points, each of them being feasible and different from zero. Let x^* be any of these limit points and let $\{x^k\}_{K\subseteq\mathbb{N}}$ be a subsequence converging to x^*. In view of Proposition 4.5, x^* is also a limit point of $\{x^{k+1}\}_K$ and of $\{x^{k-1}\}_K$. Namely,

$$\lim_{\substack{k\to\infty\\ k\in K}} x^k = \lim_{\substack{k\to\infty\\ k\in K}} x^{k+1} = \lim_{\substack{k\to\infty\\ k\in K}} x^{k-1} = x^* \neq 0. \tag{23}$$

Using Proposition 4.3, without loss of generality we can assume that Strategy AC2CD is used for all $k \in K$. (In particular, if Strategy AC2CD is used for infinitely many $k \in K$, we can simply discard from $\{x^k\}_K$ the indices k where Strategy AC2CD is not used. On the contrary, if $\bar k$ exists such that Strategy MVP is used for all $k \ge \bar k$, $k \in K$, then we can consider the subsequence $\{x^{k+1}\}_K$ instead. This subsequence still converges to x^* by (23) and, for all sufficiently large $k \in K$, Strategy AC2CD is used for all $k+1$ in view of Proposition 4.3.)

Since the set of variable indices is finite, for all $k \in K$ we can assume that $\mathcal{A}^k$, $\mathcal{N}^k$, $j(k)$ and p_i^k are the same, that is,

$$\mathcal{A}^k = \mathcal{A}, \qquad \mathcal{N}^k = \mathcal{N}, \qquad j(k) = \bar{\jmath}, \qquad p_i^k = p_i, \quad \forall i \in \{1, \dots, |\mathcal{N}|\}$$

(passing into a further subsequence if necessary) and, by point (ii) of Lemma 4.2, that

$$\lim_{\substack{k \to \infty \\ k \in K}} z^{k,i} = \bar{z}^i, \quad \forall i \in \{1, \dots, |\mathcal{N}| + 1\} \tag{24}$$

(passing again into a further subsequence if necessary).

By continuity of f, we can take the limits in (17) and, using (24), for all $i \in \{1, \dots, |\mathcal{N}|\}$ we have

$$\begin{aligned} &\bar{z}^{i+1} = \bar{z}^i + \xi^*(e_{p_i} - e_{\bar{\jmath}}), \\ &\xi^* \in \operatorname*{Argmin}_{\xi \in \mathbb{R}} \{f(\bar{z}^i + \xi(e_{p_i} - e_{\bar{\jmath}}))\}. \end{aligned} \tag{25}$$

Now, we show that

$$\bar{z}^i = x^*, \quad \forall i \in \{1, \dots, |\mathcal{N}| + 1\}. \tag{26}$$

From the instructions of the algorithm we have $z^{k,1} = x^k$, so we can write $\|z^{k,i} - x^k\| \le \sum_{h=1}^{i-1} \|z^{k,h+1} - z^{k,h}\|$ for all $i \in \{1, \dots, |\mathcal{N}| + 1\}$. Using (19b), it follows that

$$\lim_{\substack{k \to \infty \\ k \in K}} \|z^{k,i} - x^k\| = 0, \quad \forall i \in \{1, \dots, |\mathcal{N}| + 1\}. \tag{27}$$

Since $\|z^{k,i} - x^*\| \le \|z^{k,i} - x^k\| + \|x^k - x^*\|$, from (23) and (27) we get

$$\lim_{\substack{k \to \infty \\ k \in K}} \|z^{k,i} - x^*\| = 0, \quad \forall i \in \{1, \dots, |\mathcal{N}| + 1\}.$$

Using (24), we thus obtain (26).

Therefore, from (25) and (26) it follows that

$$f(x^*) = \min_{\xi \in \mathbb{R}} f(x^* + \xi(e_i - e_{\bar{\jmath}})), \quad \forall i \in \mathcal{N}. \tag{28}$$

Using (23), a real number $\eta > 0$ exists such that $\|x^k\|_\infty \ge \eta/\tau$ for all sufficiently large $k \in K$, where $\tau \in (0, 1]$ is the parameter used in `AS-ZSL` such that $|x_{\bar{\jmath}}^k| \ge \tau \|x^k\|_\infty$ (see line 9 of Algorithm 1). Consequently, for all sufficiently large $k \in K$, we have $|x_{\bar{\jmath}}^k| \ge \eta$, and then,

$$x_{\bar{\jmath}}^* \neq 0. \tag{29}$$

So, using (28), (29) and Proposition 2.4, there exists $\mu^* \in \mathbb{R}$ such that

$$\begin{cases} \nabla_i \varphi(x^*) - \mu^* = \lambda, & i \in \mathcal{N} \colon x_i^* < 0, \\ \nabla_i \varphi(x^*) - \mu^* = -\lambda, & i \in \mathcal{N} \colon x_i^* > 0, \\ |\nabla_i \varphi(x^*) - \mu^*| \le \lambda, & i \in \mathcal{N} \colon x_i^* = 0. \end{cases} \tag{30}$$

Now, from the active-set estimate (11a) we observe that $i \in \mathcal{A} \Rightarrow x_i^k = 0$ for all $k \in K$. Since $\{x^k\}_K \to x^*$ from (23), it follows that

$$\mathcal{A} \subseteq \{i \colon x_i^* = 0\}. \tag{31}$$

Taking into account (30) and (31), according to point (b) of Theorem 2.2 we thus have to show that

$$|\nabla_i \varphi(x^*) - \mu^*| \le \lambda, \quad \forall i \in \mathcal{A}, \tag{32}$$

in order to prove that x^* is optimal and conclude the proof. Note that, from (30) and (31), we can write $|x_i^*|^p \bigl(\nabla_i \varphi(x^*) + \lambda \operatorname{sign}(x_i^*)\bigr) = |x_i^*|^p \mu^*$, $i = 1, \dots, n$. This implies that

$$\sum_{i=1}^n |x_i^*|^p \bigl(\nabla_i \varphi(x^*) + \lambda \operatorname{sign}(x_i^*)\bigr) = \mu^* \sum_{i=1}^n |x_i^*|^p.$$

So, using the definition of the multiplier function $\mu(x)$ given in (12), we get

$$\mu^* = \mu(x^*). \tag{33}$$

Moreover, Proposition 4.3 ensures that, for sufficiently large $k \in K$, Strategy MVP is used at x^{k-1}, implying that $\pi^k = \pi^{k-1} = \nabla \varphi(x^{k-1}) - \mu(x^{k-1})e$ (see lines 8 and 4 of Algorithm 1). Therefore, from the active-set estimate (11a), for all sufficiently large $k \in K$ we can write

$$\mathcal{A} = \{i \colon x_i^k = 0,\ |\nabla_i \varphi(x^{k-1}) - \mu(x^{k-1})| \le \lambda\}.$$

Since $\{x^k\}_K$ and $\{x^{k-1}\}_K$ converge to x^* from (23), using the continuity of $\nabla \varphi$, the continuity of the multiplier function $\mu(x)$ and (33), we can take the limits for $k \to \infty$, $k \in K$, and we finally get (32). □

5 Numerical results

In this section, we report the numerical results obtained on synthetic and real datasets with compositional data. We implemented `AS-ZSL` in C++, using a MEX file to call the algorithm from Matlab. The `AS-ZSL` software is available at `https://github.com/acristofari/as-zsl`.

In our experiments, we use $p = 1$ for the multiplier functions defined in (12) and $\tau = 1$. Moreover, θ is set to 10^{-2} at the beginning of the algorithm and is gradually decreased to 10^{-6}. All tests were run on an Intel(R) Core(TM) i7-9700 with 16 GB RAM memory.

We compared `AS-ZSL` with the following algorithms:

- `compCL` [32], which solves (1) by the method of multiplier minimizing the augmented Lagrangian function by cyclic coordinate descent. The code was written in C++ and called from R. It was downloaded from `https://cran.r-project.org/package=Compack` as part of the R package Compack.

- `QP` [20], which solves the quadratic reformulation (3). Since $n > m$, a ridge term $10^{-4}\|x\|^2$ was added to the original objective function, as suggested in [20]. The code was downloaded from `https://github.com/Hua-Zhou/SparseReg`, it builds the problem via Matlab and uses the Gurobi Optimizer (version 9.5) [24] for the minimization.

- `zeroSum` [3], which uses an extension of the random coordinate descent method to solve (1). The code was written in C++ and called from R. It was downloaded from `https://github.com/rehbergT/zeroSum` as part of the R package zeroSum.

These algorithms were run with their default parameters and options, except those specified above. We observe that both `compCL` and `zeroSum` use a (block) coordinate descent approach and were specifically designed for regression problems with zero-sum constraint, while `QP` uses the default algorithm implemented in Gurobi for quadratic programs, i.e., the barrier algorithm.

The results obtained from the comparisons are described in Subsection 5.1 and 5.2. Finally, in Subsection 5.3 we show how a warm start strategy can be used in `AS-ZSL` to solve a sequence of problems with decreasing regularization parameters.

5.1 Synthetic datasets

In the first experiments, we generated some synthetic datasets for log-contrast model as suggested in [32], using the function *comp_Model* implemented in the Compack package. More specifically, a matrix $M \in \mathbb{R}^{m\times n}$ was first generated from a multivariate normal distribution $N(\omega, \Sigma)$. To model the presence of five major components in the composition, the vector ω has all zeros except for $\omega_i = \log(0.5n)$, $i = 1,\dots,5$. The matrix Σ has $0.5^{|i-j|}$ in position (i,j). Then, the log-contrast model (2) was obtained by setting the matrix A such that $A_{ij} = \log(Z_{ij})$ in position (i,j), with

$$Z_{ij} = \frac{e^{M_{ij}}}{\displaystyle\sum_{h=1}^{n} e^{M_{ih}}}, \quad i = 1,\dots,m, \quad j = 1,\dots,n,$$

the vector of regression coefficients $x = (1, -0.8, 0.6, 0, 0, -1.5, -0.5, 1.2, 0, \dots, 0)^T$ and the noise terms in ε were generated from a normal distribution $N(0, 0.5^2)$. We see that x has six non-zero coefficients, with three of them being among the five major components.

We considered problems with dimensions $m = 2000$ and $n \in \{m, 2m, 5m\}$. For every pair (m,n) we generated 10 different datasets and, for each of them, we used $\lambda \in \{\lambda_1,\dots,\lambda_5\}$ such that $\lambda_1,\dots,\lambda_5$ are logarithmically equally spaced between 10^{λ_1} and 10^{λ_5}, where

$$\lambda_1 = 0.95\lambda_{\max}, \quad \lambda_5 = 10^{-3}\lambda_{\max}$$

and $\lambda_{\max}$ is such that $x^* = 0$ is an optimal solution of problem (1) if and only if $\lambda \geq \lambda_{\max}$. We can easily compute $\lambda_{\max}$ from point (c) of Theorem 2.2 (it also follows from Corollary 1 of [28]):

$$\lambda_{\max} = \frac{\displaystyle\max_{j=1,\dots,n} [(A^T)y]_j - \min_{i=1\dots,n} [(A^T)y]_i}{2}.$$

In Table 1, we show the results obtained with the different problem dimensions. In particular, for each considered λ, we report the average values over the 10 runs in terms of final objective value and CPU time. We see that `AS-ZSL` always took less 2 seconds on average to solve all the considered problems, being much faster than the other methods

Table 1: Results on synthetic datasets for log-contrast models with six non-zero regression coefficients. The final objective value is indicated by f^*, while the CPU time in seconds is indicated by *time*.

$m = 2000$, $n = 2000$

	λ_1		λ_2		λ_3		λ_4		λ_5	
	f^*	time	f^*	time	f^*	time	f^*	time	f^*	time
`AS-ZSL`	4.70e+04	0.02	1.61e+04	0.03	4.71e+03	0.04	1.59e+03	0.20	5.21e+02	0.38
`compCL`	4.70e+04	2.10	1.61e+04	1.44	4.71e+03	1.44	1.59e+03	5.02	5.37e+02	5.89
`QP`	4.70e+04	7.38	1.61e+04	7.06	4.71e+03	6.40	1.59e+03	7.47	5.21e+02	7.14
`zeroSum`	4.71e+04	2.53	4.71e+04	2.54	2.79e+04	2.55	4.61e+03	2.59	2.01e+03	3.31

$m = 2000$, $n = 4000$

	λ_1		λ_2		λ_3		λ_4		λ_5	
	f^*	time	f^*	time	f^*	time	f^*	time	f^*	time
`AS-ZSL`	5.41e+04	0.04	1.84e+04	0.09	5.19e+03	0.09	1.79e+03	0.36	5.66e+02	1.22
`compCL`	5.41e+04	4.30	1.84e+04	2.77	5.19e+03	2.82	1.79e+03	9.98	5.85e+02	11.49
`QP`	5.41e+04	104.77	1.84e+04	96.41	5.19e+03	80.44	1.79e+03	105.43	5.66e+02	97.74
`zeroSum`	5.42e+04	9.97	5.42e+04	9.93	3.85e+04	10.03	5.51e+03	10.02	2.16e+03	11.47

$m = 2000$, $n = 10000$

	λ_1		λ_2		λ_3		λ_4		λ_5	
	f^*	time	f^*	time	f^*	time	f^*	time	f^*	time
`AS-ZSL`	6.50e+04	0.10	2.18e+04	0.19	5.81e+03	0.24	2.04e+03	0.79	6.30e+02	1.64
`compCL`	6.50e+04	11.30	2.18e+04	6.86	5.81e+03	7.08	2.04e+03	26.21	6.54e+02	29.84
`QP`	6.50e+04	3239.67	2.18e+04	2537.57	5.81e+03	2364.73	2.04e+03	2738.23	6.30e+02	2745.07
`zeroSum`	6.51e+04	61.90	6.51e+04	61.60	6.45e+04	61.92	7.22e+03	61.50	2.28e+03	63.73

and also achieving the lowest objective function value. In particular, `AS-ZSL` is one or two orders of magnitude faster than the other coordinate descent based methods, i.e., `compCL` and `zeroSum`.

Next, we used the same datasets described above, but with the vector of regression coefficients x containing 5% of randomly chosen non-zero entries, which were generated from a uniform distribution in $(-1, 1)$.

The results are shown in Table 2. Also in this case, we see that `AS-ZSL` achieves the lowest objective function value and is the fastest method, except for the largest problems with λ_5, where `compCL` took less time, but it returned a higher objective function value.

5.2 Real datasets

Now we show the results obtained on real microbiome data, which are generally regarded as compositional in the literature (see, e.g., [9, 22, 34, 41]). We used three datasets considered in [37], downloaded from `https://github.com/nphdang/DeepCoDA` and containing data from [47] suitably adjusted to deal with log-contrast model by proper replacement of the zero features (which are not allowed in such models). The considered datasets are for binary classification (then, responses are in $\{0, 1\}$) and are described in Table 3.

Table 2: Results on synthetic datasets for log-contrast models with 5% of non-zero regression coefficients. The final objective value is indicated by f^*, while the CPU time in seconds is indicated by *time*.

$m = 2000$, $n = 2000$

	λ_1		λ_2		λ_3		λ_4		λ_5	
	f^*	time	f^*	time	f^*	time	f^*	time	f^*	time
AS-ZSL	3.46e+04	0.03	1.85e+04	0.12	6.03e+03	0.27	1.43e+03	0.55	2.71e+02	1.48
compCL	3.46e+04	4.18	1.94e+04	10.21	8.39e+03	10.64	3.40e+03	9.83	1.52e+03	9.16
QP	3.46e+04	6.85	1.85e+04	6.74	6.03e+03	9.69	1.43e+03	6.50	2.71e+02	6.25
zeroSum	3.46e+04	2.43	2.02e+04	3.08	7.71e+03	3.95	1.49e+03	5.18	2.73e+02	40.79

$m = 2000$, $n = 4000$

	λ_1		λ_2		λ_3		λ_4		λ_5	
	f^*	time	f^*	time	f^*	time	f^*	time	f^*	time
AS-ZSL	8.52e+04	0.05	5.36e+04	0.31	3.22e+04	0.69	1.14e+04	2.20	2.33e+03	5.84
compCL	8.52e+04	17.26	5.51e+04	19.75	3.50e+04	20.39	1.48e+04	18.10	4.89e+03	17.81
QP	8.52e+04	96.86	5.36e+04	103.67	3.22e+04	95.91	1.14e+04	99.98	2.33e+03	96.76
zeroSum	8.52e+04	9.78	6.75e+04	12.09	4.45e+04	16.19	1.19e+04	22.96	2.35e+03	149.59

$m = 2000$, $n = 10000$

	λ_1		λ_2		λ_3		λ_4		λ_5	
	f^*	time	f^*	time	f^*	time	f^*	time	f^*	time
AS-ZSL	1.73e+05	0.15	1.18e+05	0.91	4.85e+04	3.52	1.40e+04	18.53	2.86e+03	95.71
compCL	1.73e+05	34.30	1.21e+05	46.58	5.41e+04	53.06	2.01e+04	40.07	5.68e+03	30.30
QP	1.73e+05	2622.73	1.18e+05	2890.54	4.85e+04	2963.40	1.40e+04	3216.94	2.86e+03	3586.99
zeroSum	1.73e+05	61.94	1.23e+05	92.17	5.47e+04	206.69	1.52e+04	700.20	3.24e+03	1930.67

For each dataset, first we applied a log transformation and then we chose λ by a 5-fold cross validation. The final results are shown in Table 4. We see that AS-ZSL took less than 1 second on all the problems, still being the fastest method and achieving the lowest objective function value.

5.3 Optimizing over a grid of regularization parameters

In the previous experiments, we have shown the performances of AS-ZSL on several instances of problems (1) for a specific value of the regularization parameter λ. Now, we want to analyze a simple *warm start* strategy for AS-ZSL to solve a sequence of problems with decreasing regularization parameters. This can be useful in practice when a suitable value of λ is not

Table 3: Microbiome datasets from [37, 47].

Dataset	m	n	Class 1	Class 2
1	2070	3090	Gastro	Oral
2	404	3090	Stool	Tongue
3	408	3090	Subgingival	Supragingival

Table 4: Results on microbiome datasets. The final objective value is indicated by f^*, while the CPU time in seconds is indicated by *time*.

	Dataset 1		Dataset 2		Dataset 3	
	f^*	time	f^*	time	f^*	time
`AS-ZSL`	3.38e+01	0.48	2.84e+01	0.20	7.18e+01	0.08
`compCL`	3.38e+01	13.86	2.95e+01	2.61	7.18e+01	4.39
`QP`	3.38e+01	95.63	2.84e+01	57.50	7.18e+01	81.98
`zeroSum`	2.56e+02	7.27	7.35e+01	1.77	1.05e+02	0.97

Table 5: Comparison of CPU time required by `AS-ZSL` with and without warm start over 10 values of λ. All results are in seconds.

	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6	λ_7	λ_8	λ_9	λ_{10}
time with warm start	0.14	0.46	0.99	1.77	2.95	4.87	7.83	12.34	15.60	19.45
time without warm start	0.14	0.44	0.80	1.29	2.79	5.24	10.94	24.69	52.16	104.07
cumulative time with warm start	0.14	0.60	1.59	3.36	6.31	11.18	19.01	31.35	46.95	66.40
cumulative time without warm start	0.14	0.58	1.38	2.67	5.46	10.69	21.63	46.32	98.48	202.55

known in advance and a parameter selection procedure must be carried out.

We generated 10 synthetic datasets as explained in Subsection 5.1, with $m = 2000$, $n = 10000$ and 5% of non-zero entries in the vector of regression coefficients. For each dataset, we considered 10 different parameters $\lambda_1, \dots, \lambda_{10}$ logarithmically equally spaced between 10^{λ_1} and $10^{\lambda_{10}}$, where $\lambda_1 = 0.95\lambda_{\max}$, $\lambda_{10} = 10^{-3}\lambda_{\max}$ and $\lambda_{\max}$ was computed as explained in Subsection 5.1.

For $\lambda = \lambda_i$, $i = 2, \dots, 10$, the warm start strategy simply consists in setting the starting point of `AS-ZSL` as the optimal solution computed with $\lambda = \lambda_{i-1}$.

In Table 5, we report the average results obtained with and without warm start. From the third and the fourth row of the table, we observe that the warm start strategy allowed us to complete the whole optimization process in 66 seconds, while it took more than 200 seconds without warm start. From the second and the third row of the table, we also note that running `AS-ZSL` with one small λ took more than running `AS-ZSL` several times using decreasing regularization parameters with warm start. For example, `AS-ZSL` took 104 seconds for $\lambda = \lambda_{10}$ without warm start, but it took a total of 66 seconds when it was run ten times with $\lambda = \lambda_1, \dots, \lambda_{10}$ using the warm start strategy. This suggests that the warm start strategy might be used to further speed up the algorithm when problem (1) must be solved with small values of λ.

6 Conclusions

In this paper we proposed `AS-ZSL`, a 2-coordinate descent method with active-set estimate to solve lasso problems with zero-sum constraint. At every iteration, `AS-ZSL` chooses between two strategies: Strategy MVP uses the whole gradient of the smooth term of the objective function and is more accurate in the active-set estimate, while Strategy AC2CD only needs partial derivatives and is computationally more efficient. A suitable test is used to choose between the two strategies, considering the progress in the objective function. A theoretical analysis was carried out, showing global convergence of AS-ZSL to optimal solutions. We performed numerical experiments on synthetic and real datasets, showing the effectiveness of the proposed method compared to other algorithms from the literature. In particular, the good numerical results are due both to the identification of the zero variables carried out by the proposed active-set estimate and to the possibility of choosing, at each iteration, between Strategy MVP and Strategy AC2CD, in order to balance accuracy and computational efficiency.

We finally outlined a warm start strategy for AS-ZSL to solve a sequence of problems with decreasing regularization parameters, which may be useful when a parameter selection procedure must be carried out and many problems with different regularization parameters must be solved.

Possible directions for future research might include the extension of these results to a more general class of problems, for example considering other loss functions, other regularization terms and other types of constraints.

Appendix A. The subproblem

According to (15) and (17), every variable update in AS-ZSL requires the resolution of a subproblem, both when using Strategy MVP and when using Strategy AC2CD. As highlighted in Remark 2, each subproblem consists in the minimization of f along a direction of the form $\pm(e_i - e_j)$. In the next proposition, we give an equivalent expression of f along $\pm(e_i - e_j)$ as an univariate function, which will be useful to compute the minimizer.

Proposition A.1. *Let $\bar{x}$ be any feasible point of problem* (1). *For any $i \neq j$, we have*

$$f(\bar{x} + \xi(e_i - e_j)) = f^{i,j}_{\bar{x}}(\bar{x}_i + \xi), \quad \forall \xi \in \mathbb{R},$$

where $f^{i,j}_{\bar{x}} : \mathbb{R} \to \mathbb{R}$ is the function defined as follows:

$$f^{i,j}_{\bar{x}}(u) = \frac{1}{2}\alpha u^2 - \beta u + \lambda(|u| + |u - \bar{x}_i - \bar{x}_j|) + c,$$

with

$$\begin{aligned} \alpha &= \|A^i - A^j\|^2, \\ \beta &= \alpha \bar{x}_i - \nabla_i \varphi(\bar{x}) + \nabla_j \varphi(\bar{x}), \\ c &= \frac{1}{2}\|y - A\bar{x} + (A^i - A^j)\bar{x}_i\|^2 + \lambda \sum_{t \neq i,j} |\bar{x}_t|. \end{aligned}$$

Proof. First, let us write the function f as follows:

$$f(x) = \frac{1}{2}\sum_{h=1}^{m}\biggl(A_{hi}x_i + A_{hj}x_j + \sum_{t \neq i,j} A_{ht}x_t - y_h\biggr)^2 + \lambda\biggl(|x_i| + |x_j| + \sum_{t \neq i,j} |x_t|\biggr).$$

Now, choose any $\xi \in \mathbb{R}$ and let $u = \bar{x} + \xi(e_i - e_j)$. To prove the desired result, we have to show that

$$f(u) = f_{\bar{x}}^{i,j}(u_i) = \frac{1}{2}\alpha u_i^2 - \beta^T u_i + \lambda(|u_i| + |u_i - \bar{x}_i - \bar{x}_j|) + c. \tag{34}$$

Clearly, $u_t = \bar{x}_t$ for all $t \neq i, j$. Since $e^T \bar{x} = 0$ from the feasibility of $\bar{x}$, it follows that $e^T u = 0$ and $u_j = -\sum_{t \neq i,j} u_t - u_i = -\sum_{t \neq i,j} \bar{x}_t - u_i = \bar{x}_i + \bar{x}_j - u_i$. Then,

$$\begin{aligned} f(u) =& \frac{1}{2}\sum_{h=1}^{m}\biggl((A_{hi} - A_{hj})u_i + \sum_{t \neq i,j}(A_{ht} - A_{hj})\bar{x}_t - y_h\biggr)^2 + \\ & + \lambda\biggl(|u_i| + |u_i - \bar{x}_i - \bar{x}_j| + \sum_{t \neq i,j}|\bar{x}_t|\biggr). \end{aligned}$$

Now, let us define the vector $\rho \in \mathbb{R}^m$ as

$$\rho_h = y_h - \sum_{t \neq i,j}(A_{ht} - A_{hj})\bar{x}_t, \quad h = 1, \dots, m,$$

so that

$$\begin{aligned} f(u) &= \frac{1}{2}\|(A^i - A^j)u_i - \rho\|^2 + \lambda\biggl(|u_i| + |u_i - \bar{x}_i - \bar{x}_j| + \sum_{t \neq i,j}|\bar{x}_t|\biggr) \\ &= \frac{1}{2}\alpha^2 u_i^2 - \rho^T(A^i - A^j)u_i + \frac{1}{2}\|\rho\|^2 + \lambda\biggl(|u_i| + |u_i - \bar{x}_i - \bar{x}_j| + \sum_{t \neq i,j}|\bar{x}_t|\biggr). \end{aligned}$$

For all $h = 1, \dots, m$, we can write

$$\begin{aligned} \rho_h &= y_h - \sum_{t \neq i,j} A_{ht}\bar{x}_t + A_{hj}\sum_{t \neq i,j}\bar{x}_t = y_h - (A\bar{x})_h + A_{hi}\bar{x}_i + A_{hj}\bar{x}_j + A_{hj}\sum_{t \neq i,j}\bar{x}_t, \\ &= y_h - (A\bar{x})_h + (A_{hi} - A_{hj})\bar{x}_i, \end{aligned}$$

where the last inequality follows from the fact that $\bar{x}_j = -\sum_{t \neq i,j}\bar{x}_t - \bar{x}_i$, since $e^T\bar{x} = 0$. Therefore,

$$\rho = y - A\bar{x} + (A^i - A^j)\bar{x}_i \tag{35}$$

and we obtain

$$f(u) = \frac{1}{2}\alpha u_i^2 - \rho^T(A^i - A^j)u_i + \lambda\biggl(|u_i| + |u_i - \bar{x}_i - \bar{x}_j|\biggr) + c.$$

Finally, since $(A\bar{x} - y)^T A^i = \nabla_i\varphi(\bar{x})$ and $(A\bar{x} - y)^T A^j = \nabla_j\varphi(\bar{x})$, from (35) it follows that $\rho^T(A^i - A^j) = \|A^i - A^j\|^2\bar{x}_i - \nabla_i\varphi(\bar{x}) + \nabla_j\varphi(\bar{x}) = \alpha\bar{x}_i - \nabla_i\varphi(\bar{x}) + \nabla_j\varphi(\bar{x}) = \beta$, thus proving (34). □

Appendix A.1. Ensuring strict convexity

In the convergence analysis of `AS-ZSL`, we require f to be strictly convex along $\pm(e_i - e_j)$ for every index pair (i, j) selected by Strategy MVP or Strategy AC2CD (see Subsection 4.1). Using Proposition A.1, it is now easy to show that this requirement is satisfied if and only if $A^i \neq A^j$.

Corollary A.2. *Let $\bar{x}$ be a feasible point for problem* (1)*. For any $i \neq j$, we have that f is strictly convex over the line $\{\bar{x} + \xi(e_i - e_j)),\ \xi \in \mathbb{R}\}$ if and only if $A^i \neq A^j$.*

Proof. The result follows from Proposition A.1, observing that $f^{i,j}_{\bar{x}}$ is strictly convex if and only if $A^i \neq A^j$ (from the expression of α) and that any pair of distinct points x', x'' over the line $\{\bar{x} + \xi(e_i - e_j)),\ \xi \in \mathbb{R}\}$ can be expressed as $x' = \bar{x} + \xi'(e_i - e_j)$ and $x'' = \bar{x} + \xi''(e_i - e_j)$, for some $\xi' \neq \xi''$. □

When $A^i = A^j$, in the next proposition we show the variable x_i can be safely removed from the problem.

Proposition A.3. *Assume that $A^i = A^j$ for some $i \neq j$. If x^* is an optimal solution of* $\min\{f(x) \colon e^T x = 0, x_i = 0\}$*, then x^* is an optimal solution of problem* (1) *as well.*

Proof. By contradiction, assume that x^* is not an optimal solution of problem (1). Then, there exists a feasible point x' for problem (1) such that $f(x') < f(x^*)$. Now, let us define x'' as follows:

$$x''_h = \begin{cases} x'_h, & \text{if } h \in \{1, \dots, n\} \setminus \{i, j\}, \\ 0, & \text{if } h = i, \\ x'_i + x'_j, & \text{if } h = j. \end{cases}$$

Clearly, $e^T x'' = e^T x' = 0$. Since $A^i = A^j$, we have $Ax'' = Ax'$ and, using the triangular inequality, $\|x''\|_1 \leq \|x'\|_1$. It follows that $f(x'') \leq f(x') < f(x^*)$, contradicting the fact that x^* is an optimal solution of $\min\{f(x) \colon e^T x = 0, x_i = 0\}$. □

In conclusion, Proposition A.3 and Corollary A.2 suggest a simple procedure to ensure that, after a finite number of iterations, f is strictly convex along $\pm(e_i - e_j)$ for every index pair (i, j) selected by Strategy MVP or Strategy AC2CD. Namely, if we find two identical columns A^i and A^j, we can simply fix $x_i = 0$ and remove this variable from problem (1) (together with the column A^i). We note that checking if two columns A^i and A^j are identical does not require additional computational burden because, as explained below, $\|A^i - A^j\|$ must be computed anyway (in order to calculate the coefficients α and β appearing in Proposition A.1).

Appendix A.2. Computing the optimal solution

Given a feasible point $\bar{x}$ of problem (1) and an index pair (i, j), with $i \neq j$, now we show how to compute

$$\begin{aligned} &\hat{x} = \bar{x} + \xi^*(e_i - e_j), \\ &\xi^* \in \operatorname*{Argmin}_{\xi \in \mathbb{R}} \{f(\bar{x} + \xi(e_i - e_j))\}. \end{aligned}$$

According to (15) and (17), a computation of this form is needed for the variable update both when using Strategy MVP and when using Strategy AC2CD. Note that $\hat{x}$ can be equivalently obtained as an optimal solution of the following problem:

$$\begin{aligned} &\min f(x) \\ &x \in \{\bar{x} + \xi(e_i - e_j),\, \xi \in \mathbb{R}\}. \end{aligned} \tag{36}$$

So, in view of Proposition A.1, we can calculate

$$u^* \in \operatorname{Argmin} f_{\bar{x}}^{i,j}(u)$$

and set

$$\hat{x}_h = \begin{cases} u^*, & \text{if } h = i, \\ \bar{x}_h, & \text{if } h \neq i, j, \\ -\sum_{t \neq j} \hat{x}_t, & \text{if } h = j. \end{cases}$$

To compute u^*, let us first recall that, from Proposition A.1, we have

$$f_{\bar{x}}^{i,j}(u) = \frac{1}{2}\alpha u^2 - \beta u + \lambda(|u| + |u - \bar{x}_i - \bar{x}_j|) + c, \tag{37}$$

where $\alpha = \|A^i - A^j\|^2$, $\beta = \alpha \bar{x}_i - \nabla_i \varphi(\bar{x}) + \nabla_j \varphi(\bar{x})$ and c is a constant. Since $f_{\bar{x}}^{i,j}$ is coercive, then it has a minimizer. If $\alpha = 0$ (i.e., if $A^i = A^j$), we explained above that we can simply fix $x_i = 0$ and remove this variable from the problem. So, here we only focus on $\alpha > 0$. In this case, observe that $f_{\bar{x}}^{i,j}$ is strictly convex, has a unique minimizer u^* and we can write

$$f_{\bar{x}}^{i,j}(u) = \begin{cases} \frac{1}{2}\alpha u^2 - \beta u + 2\lambda u - \lambda(\bar{x}_i + \bar{x}_j) + c, & \text{if } u \geq 0 \text{ and } u \geq \bar{x}_i - \bar{x}_j, \\ \frac{1}{2}\alpha u^2 - \beta u - 2\lambda u + \lambda(\bar{x}_i + \bar{x}_j) + c, & \text{if } u \leq 0 \text{ and } u \leq \bar{x}_i - \bar{x}_j, \\ \frac{1}{2}\alpha u^2 - \beta u + \lambda(\bar{x}_i + \bar{x}_j) + c, & \text{if } u \geq 0 \text{ and } u < \bar{x}_i - \bar{x}_j, \\ \frac{1}{2}\alpha u^2 - \beta u - \lambda(\bar{x}_i + \bar{x}_j) + c, & \text{if } u \leq 0 \text{ and } u > \bar{x}_i - \bar{x}_j. \end{cases}$$

Hence, we can first seek a stationary point of $f_{\bar{x}}^{i,j}$ where the function is differentiable, i.e., in $\mathbb{R} \setminus \{0, \bar{x}_i + \bar{x}_j\}$. If such a stationary point exists, then it will be the desired minimizer u^*. Otherwise, u^* will be a point of non-differentiability, that is, either 0 or $\bar{x}_i + \bar{x}_j$. This procedure is reported in Algorithm 2.

Note that, in addition to $\nabla_i \varphi(\bar{x})$ and $\nabla_j \varphi(\bar{x})$, in Algorithm 2 we only have to compute $\|A^i - A^j\|^2$ to get α and β, with a cost of $\mathcal{O}(m)$ operations.

Algorithm 2 `to compute the minimizer` u^* `of` (37)

```
*seek a stationary point*
Set $\bar{u} = \frac{\beta - 2\lambda}{\alpha}$
**If** $\bar{u} > \max\{\bar{x}_i + \bar{x}_j, 0\}$
    Set $u^* = \bar{u}$ and EXIT (*stationary point found*)
**Else**
    Set $\bar{u} = \frac{\beta + 2\lambda}{\alpha}$
    **If** $\bar{u} < \min\{\bar{x}_i + \bar{x}_j, 0\}$
        Set $u^* = \bar{u}$ and EXIT (*stationary point found*)
    **Else**
        Set $\bar{u} = \frac{\beta}{\alpha}$
        **If** $\bar{u}(\bar{u} - \bar{x}_i - \bar{x}_j) < 0$
            Set $u^* = \bar{u}$ and EXIT (*stationary point found*)
        **End if**
    **End if**
**End if**

*stationary point not found, the minimizer is a point of non-differentiability*
**If** $f_{\bar{x}}^{i,j}(0) \leq f_{\bar{x}}^{i,j}(\bar{x}_i + \bar{x}_j)$, then set $u^* = 0$
**Else**, set $u^* = \bar{x}_i + \bar{x}_j$
```

References

[1] J. Aitchison. The statistical analysis of compositional data. *Journal of the Royal Statistical Society: Series B (Methodological)*, 44(2):139–160, 1982.

[2] J. Aitchison and J. Bacon-Shone. Log contrast models for experiments with mixtures. *Biometrika*, 71(2):323–330, 1984.

[3] M. Altenbuchinger, T. Rehberg, H. Zacharias, F. Stämmler, K. Dettmer, D. Weber, A. Hiergeist, A. Gessner, E. Holler, P. J. Oefner, and R. Spang. Reference point insensitive molecular data analysis. *Bioinformatics*, 33(2):219–226, 2017.

[4] M. Andretta, E. G. Birgin, and J. M. Martínez. Practical active-set Euclidian trust-region method with spectral projected gradients for bound-constrained minimization. *Optimization*, 54(3):305–325, 2005.

[5] A. Beck. The 2-coordinate descent method for solving double-sided simplex constrained minimization problems. *Journal of Optimization Theory and Applications*, 162(3):892–919, 2014.

[6] E. G. Birgin and J. M. Martínez. Large-scale active-set box-constrained optimization method with spectral projected gradients. *Computational Optimization and Applications*, 23(1):101–125, 2002.

[7] R. H. Byrd, G. M. Chin, J. Nocedal, and F. Oztoprak. A family of second-order methods for convex ℓ_1-regularized optimization. *Mathematical Programming*, 159(1):435–467, 2016.

[8] C.-C. Chang and C.-J. Lin. LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):1–27, 2011.

[9] P. L. Combettes and C. L. Müller. Regression models for compositional data: General log-contrast formulations, proximal optimization, and microbiome data applications. *Statistics in Biosciences*, 13(2):217–242, 2021.

[10] A. Cristofari. An almost cyclic 2-coordinate descent method for singly linearly constrained problems. *Computational Optimization and Applications*, 73(2):411–452, 2019.

[11] A. Cristofari. Active-set identification with complexity guarantees of an almost cyclic 2-coordinate descent method with Armijo line search. *SIAM Journal on Optimization*, 32(2):739–764, 2022.

[12] A. Cristofari, M. De Santis, S. Lucidi, and F. Rinaldi. A Two-Stage Active-Set Algorithm for Bound-Constrained Optimization. *Journal of Optimization Theory and Applications*, 172(2):369–401, 2017.

[13] A. Cristofari, M. De Santis, S. Lucidi, and F. Rinaldi. An active-set algorithmic framework for non-convex optimization problems over the simplex. *Computational Optimization and Applications*, 77:57–89, 2020.

[14] A. Cristofari, M. De Santis, S. Lucidi, and F. Rinaldi. Minimization over the ℓ_1-ball using an active-set non-monotone projected gradient. *Computational Optimization and Applications*, 2022.

[15] A. Cristofari, G. Di Pillo, G. Liuzzi, and S. Lucidi. An augmented Lagrangian method exploiting an active-set strategy and second-order information. *Journal of Optimization Theory and Applications*, 2022.

[16] M. De Santis, S. Lucidi, and F. Rinaldi. A Fast Active Set Block Coordinate Descent

Algorithm for ℓ_1-Regularized Least Squares. *SIAM Journal on Optimization*, 26(1): 781–809, 2016.

[17] Z. Deng, M.-C. Yue, and A. M.-C. So. An Efficient Augmented Lagrangian-Based Method for Linear Equality-Constrained Lasso. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5760–5764. IEEE, 2020.

[18] F. Facchinei and S. Lucidi. Quadratically and superlinearly convergent algorithms for the solution of inequality constrained minimization problems. *Journal of Optimization Theory and Applications*, 85(2):265–289, 1995.

[19] F. Facchinei, J. Júdice, and J. Soares. An active set Newton algorithm for large-scale nonlinear programs with box constraints. *SIAM Journal on Optimization*, 8(1):158–186, 1998.

[20] B. R. Gaines, J. Kim, and H. Zhou. Algorithms for fitting the constrained lasso. *Journal of Computational and Graphical Statistics*, 27(4):861–871, 2018.

[21] L. E. Ghaoui, V. Viallon, and T. Rabbani. Safe feature elimination for the lasso and sparse supervised learning problems. *Pacific Journal of Optimization*, 8(4):667–698, 2010.

[22] G. B. Gloor, J. M. Macklaim, V. Pawlowsky-Glahn, and J. J. Egozcue. Microbiome datasets are compositional: and this is not optional. *Frontiers in Microbiology*, 8:2224, 2017.

[23] L. Grippo and M. Sciandrone. On the convergence of the block nonlinear Gauss–Seidel method under convex constraints. *Operations Research Letters*, 26(3):127–136, 2000.

[24] Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2021. URL `https://www.gurobi.com`.

[25] W. W. Hager and D. A. Tarzanagh. A Newton-type active set method for nonlinear optimization with polyhedral constraints. *arXiv preprint arXiv:2011.01201*, 2020.

[26] W. W. Hager and H. Zhang. A new active set algorithm for box constrained optimization. *SIAM Journal on Optimization*, 17(2):526–557, 2006.

[27] C.-J. Hsieh, K.-W. Chang, C.-J. Lin, S. S. Keerthi, and S. Sundararajan. A dual coordinate descent method for large-scale linear SVM. In *Proceedings of the 25th International Conference on Machine Learning*, pages 408–415, 2008.

[28] J.-J. Jeon, Y. Kim, S. Won, and H. Choi. Primal path algorithm for compositional data analysis. *Computational Statistics & Data Analysis*, 148:106958, 2020.

[29] T. Joachims. Making large-scale support vector machine learning practical. In B. Schölkopf, C. J. Burges, and A. J. Smola, editors, *Advances in Kernel Methods – Support Vector Learning, B*, pages 169–184. MIT Press, 1999.

[30] N. Keskar, J. Nocedal, F. Öztoprak, and A. Waechter. A second-order method for convex ℓ_1-regularized optimization with active-set prediction. *Optimization Methods and Software*, 31(3):605–621, 2016.

[31] C.-J. Lin. On the convergence of the decomposition method for support vector machines. *IEEE Transactions on Neural Networks*, 12(6):1288–1298, 2001.

[32] W. Lin, P. Shi, R. Feng, and H. Li. Variable selection in regression with compositional covariates. *Biometrika*, 101(4):785–797, 2014.

[33] Z.-Q. Luo and P. Tseng. On the convergence of the coordinate descent method for

convex differentiable minimization. *Journal of Optimization Theory and Applications*, 72(1):7–35, 1992.

[34] A. Mishra and C. L. Müller. Robust regression with compositional covariates. *Computational Statistics & Data Analysis*, 165:107315, 2022.

[35] L. Palagi and M. Sciandrone. On the convergence of a modified version of SVM light algorithm. *Optimization Methods and Software*, 20(2-3):317–334, 2005.

[36] J. C. Platt. Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines. In B. Schölkopf, C. J. Burges, and A. J. Smola, editors, *Advances in Kernel Methods – Support Vector Learning*, pages 185–208. MIT Press, 1998.

[37] T. Quinn, D. Nguyen, S. Rana, S. Gupta, and S. Venkatesh. Deepcoda: personalized interpretability for compositional health data. In *International Conference on Machine Learning*, pages 7877–7886. PMLR, 2020.

[38] K. Scheinberg. An efficient implementation of an active set method for SVMs. *Journal of Machine Learning Research*, 7:2237–2257, 2006.

[39] M. Schmidt, G. Fung, and R. Rosales. Fast optimization methods for l1 regularization: A comparative study and two new approaches. In *European Conference on Machine Learning*, pages 286–297. Springer, 2007.

[40] A. L. Schwartz and E. Polak. Family of projected descent methods for optimization problems with simple bounds. *Journal of Optimization Theory and Applications*, 92(1): 1–31, 1997.

[41] P. Shi, A. Zhang, and H. Li. Regression analysis for microbiome compositional data. *The Annals of Applied Statistics*, 10(2):1019–1040, 2016.

[42] S. Solntsev, J. Nocedal, and R. H. Byrd. An algorithm for quadratic ℓ_1-regularized optimization with a flexible active-set strategy. *Optimization Methods and Software*, 30 (6):1213–1237, 2015.

[43] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

[44] R. Tibshirani, J. Bien, J. Friedman, T. Hastie, N. Simon, J. Taylor, and R. J. Tibshirani. Strong rules for discarding predictors in lasso-type problems. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 74(2):245–266, 2012.

[45] R. J. Tibshirani and J. Taylor. The solution path of the generalized lasso. *The Annals of Statistics*, 39(3):1335–1371, 2011.

[46] P. Tseng. Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of Optimization Theory and Applications*, 109(3):475–494, 2001.

[47] P. Vangay, B. M. Hillmann, and D. Knights. Microbiome Learning Repo (ML Repo): A public repository of microbiome regression and classification tasks. *Gigascience*, 8(5), 2019.

[48] J. Wang, J. Zhou, P. Wonka, and J. Ye. Lasso screening rules via dual polytope projection. *Advances in Neural Information Processing Systems*, 26, 2013.

[49] Z. Wen, W. Yin, D. Goldfarb, and Y. Zhang. A fast algorithm for sparse reconstruction based on shrinkage, subspace optimization, and continuation. *SIAM Journal on Scientific Computing*, 32(4):1832–1857, 2010.

[50] Z. Wen, W. Yin, H. Zhang, and D. Goldfarb. On the convergence of an active-set method for ℓ_1 minimization. *Optimization Methods and Software*, 27(6):1127–1146, 2012.

[51] Z. Xiang, H. Xu, and P. J. Ramadge. Learning sparse representations of high dimensional data on large scale dictionaries. *Advances in Neural Information Processing Systems*, 24, 2011.

[52] Z. J. Xiang and P. J. Ramadge. Fast lasso screening tests based on correlations. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2137–2140. IEEE, 2012.

[53] G.-X. Yuan, K.-W. Chang, C.-J. Hsieh, and C.-J. Lin. A comparison of optimization methods and software for large-scale l1-regularized linear classification. *Journal of Machine Learning Research*, 11:3183–3234, 2010.