Partitioned Least Squares

Partitioned Least Squares

Roberto Esposito[*], Mattia Cerrato[†], Marco Locatelli[‡]

November 25, 2019

Abstract

Linear least squares is one of the most widely used regression methods among scientists in many fields. The simplicity of the model allows this method to be used when data is scarce and it is usually appealing to practitioners that need to gather some insight into the problem by inspecting the values of the learnt parameters. In this paper we propose a variant of the linear least squares model that allows practitioners to partition the input features into groups of variables that they require to contribute similarly to the final result. We formally show that the new formulation is not convex and provide two alternative methods to deal with the problem: one non-exact method based on an alternating least squares approach; and one exact method based on a reformulation of the problem using an exponential number of sub-problems whose minimum is guaranteed to be the optimal solution. We formally show the correctness of the exact method and also compare the two solutions showing that the exact solution provides better results in a fraction of the time required by the alternating least squares solution (assuming that the number of partitions is small).

1 Introduction

Linear regression models are among the most extensively employed statistical methods in science and industry alike. Their simplicity, ease of use and performance in low-data regimes enables their usage in various prediction tasks. As the number of observations usually exceeds the number of variables, a practitioner has to resort to approximating the solution of an overdetermined system. Least squares approximation benefits from a closed-form solution and might be the de-facto standard in linear regression analysis. Among the benefits of linear

[*]Dipartimento di Informatica, Università di Torino, 10149 Torino, Italy, roberto.esposito@unito.it

[†]Dipartimento di Informatica, Università di Torino, 10149 Torino, Italy, mattia.cerrato@unito.it

[‡]Dipartimento di Ingegneria e Architettura, Università di Parma, 43124 Parma, Italy, marco.locatelli@unipr.it

regression models is the possibility of easily interpreting how much each variate is contributing to the approximation of the dependent variable by means of observing the magnitudes and signs of the associated parameters.

In some application domains, partitioning the variables in non-overlapping subsets is beneficial either as a way to insert human knowledge into the regression analysis task or to further improve model interpretability. When considering high-dimensionality data, grouping variables together is also a natural way to make it easier to reason about the data and the regression result. As an example, consider a regression task where the dependent variable is the score achieved by students in an University or College exam. A natural way to group the dependent variables is to divide them into two groups where one contains the variables which represent a student's effort in the specific exam (hours spent studying, number of lectures attended...), while another contains the variables related to previous effort and background (number of previous exams passed, number of years spent at University or College, grade average...). As an another example, when analyzing complex chemical compounds, it is possible to group together fine-grained features to obtain a partition which refers to high-level properties of the compound (such as structural, interactive and bond-forming among others).

In this paper, we introduce a variation on the linear regression problem which allows for partitioning variables into meaningful groups. The parameters obtained by solving the problem allows one to easily assess the contribution of each group to the dependent variable as well as the importance of each element of the group.

Our contributions include a formal non-convexity proof for the new Partitioned Least Squares problem and two possible algorithms to solve it. One is based on the Alternating Least Squares algorithm, where the optimization of the parameters is iterative and can get trapped into local minima; the other is based on a reformulation of the original problem into an exponential number of subproblems, where the exponent is the cardinality K of the partition. We prove that solutions found by the second approach are globally optimal and test both algorithms on data extracted from the analysis of chemical compounds. Our experimental results show that the optimal algorithm is also faster, provided that the size of the partition is small.

While to the best of our knowledge the regression problem and the algorithms we present are novel, there has been previous work dealing with alternative formulations to the linear regression problem. Partial Least Squares Regression [8] parametrizes both the dependent and independent variables; Weighted Linear Regression minimizes the residuals' *weighted* sum of squares. Partitioned variables have also been the subject of previous work dealing with *selecting* groups of features given a partitioning. Huang et al. provide a review of such methodologies [6].

Table 1: Notation

Symbol(s)	Definition
$(\cdot)_n$	n-th component of a vector
k, K	k is the index for iterating over the K subsets belonging to the partition
m, M	m is the index for iterating over the M variables
$\mathbf{X}$	an $N \times M$ matrix containing the descriptions of the training instances
$\mathbf{A} \times \mathbf{B}$	matrix multiplication operation (we also simply write it $\mathbf{AB}$ when the notation appears clearer)
$\boldsymbol{y}$	a vector of length N containing the labels assigned to the examples in $\mathbf{X}$
$\bullet$	wildcard used in subscriptions to denote whole columns or whole rows: e.g., $\boldsymbol{X}_{\bullet,k}$ denotes the k-th column of matrix $\mathbf{X}$ and $\boldsymbol{X}_{m,\bullet}$ denotes its m-th row
$*$	denotes an optimal solution, e.g., $p^\star$ denotes the optimal solution of the PartitionedLS problem, while $p_b^\star$ denotes the optimal solution of the PartitionedLS-b problem
$\mathbf{P}$	a $M \times K$ partition matrix, $P_{m,k} \in \{0,1\}$, with $P_{m,k} = 1$ iff variable α_m belongs to the k-th element of the partition
P_k	the set of all indices in the k-th element of the partition: $\{m \vert P_{k,m} = 1\}$
$k[m]$	index of the partition element to which α_m belongs, i.e.: $k[m]$ is such that $m \in P_{k[m]}$
$\circ$	Hadamard (i.e., element-wise) product. When used to multiply a matrix by a column vector, it is intended that the columns of the matrix are each one multiplied (element-wise) by the column vector
$\oslash$	Hadamard (i.e., element-wise) division
$\succeq$	element-wise larger-than operator: $\boldsymbol{\alpha} \succeq 0$ is equivalent to $\alpha_m \geq 0$ for $m \in 1..M$

2 Model description

Let us consider the problem of inferring a linear least squares model to predict a real variable y given a vector $\boldsymbol{x} \in \mathbf{R}$. We will assume that the examples are available at learning time as an $N \times M$ matrix $\mathbf{X}$ and $N \times 1$ column vector $\boldsymbol{y}$. We will also assume that the problem is expressed in homogeneous coordinates, i.e., that $\mathbf{X}$ has an additional column containing values equal to 1, and that the intercept term of the affine function is included into the weight vector.

The standard least squares formulation for the problem at hand is to minimize the quadratic loss over the residuals, i.e.:

$$\text{minimize}_{\boldsymbol{w}} \|\mathbf{X}\boldsymbol{w} - \boldsymbol{y}\|_2^2$$

This is a problem that has the closed form solution $\boldsymbol{w} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\boldsymbol{y}$. As mentioned in Section 1, in many application contexts where M is large, the resulting model is hard to interpret. However, it is often the case that domain experts can partition the elements in the weights vector into a small number of groups and that a model built on this partition would be much easier to interpret. Then, let $\mathbf{P}$ be a "partition" matrix for the problem at hand (this is not a partition matrix in the linear algebra sense, it is simply a matrix containing the information needed to partition the features of the problem). More formally, let $\mathbf{P}$ be a $M \times K$ matrix where $P_{m,k} \in \{0, 1\}$ is 1 iff feature number m belongs to the k-th partition element. We will also write P_k to denote the set $\{m|P_{m,k} = 1\}$.

Here we introduce the Partitioned Least Square (PartitionedLS) problem, a model where we introduce K additional variables and try to express the whole regression problem in terms of these new variables (and in terms of how the original variables contribute to the predictions made using them). The simplest way to describe the new model is to consider its regression function:

$$f(\mathbf{X}) = \left(\sum_{k=1}^{K} \beta_k \sum_{m \in P_k} \alpha_m x_{n,m} + t \right)_n \tag{1}$$

where $(\cdot)_n$ denotes the n-th component of the vector being built. The first summation is over the K sets in the partition that domain experts have identified as interesting, while the second one iterates over all variables in that set. We note that the m-th α weight contributes to the k-th element of the partition only if it belongs to it. As we shall see, we require that all α values are not smaller than 0 and that $\forall k : \sum_{m \in P_k} \alpha_m = 1$. Consequently, the expression returns a vector of predictions calculated in terms of two sets of weights: the β weights, which are meant to capture the magnitude and the sign of the contribution of the k-th element of the partition, and the α weights, which are meant to capture how each feature in the k-th set contributes to it. We note that the α weight vector is of the same length as the vector $\boldsymbol{w}$ in the least squares formulation. Despite this similarity, we prefer to use a different symbol because the interpretation of (and the constraints on) the α weights are different with respect to the w weights.

It is easy to verify that the definition of f in (1) can be rewritten in matrix notation as:

$$f(\mathbf{X}) = \left(\sum_{k=1}^{K} \beta_k \sum_{m} P_{m,k} \alpha_m x_{n,m} + t \right)_n = \mathbf{X} \times (\mathbf{P} \circ \boldsymbol{\alpha}) \times \boldsymbol{\beta} + t \qquad (2)$$

where $\circ$ is the Hadamard product extended to handle column-wise products. More formally, if $\mathbf{Z}$ is a $A \times B$ matrix, $\mathbf{1}$ is a B dimensional vector with all entries equal to 1, and $\boldsymbol{a}$ is a column vector of length A, then $\mathbf{Z} \circ \boldsymbol{a} \triangleq \mathbf{Z} \circ (\boldsymbol{a} \times \mathbf{1}^T)$; where the $\circ$ symbol on the right hand side of the definition is the standard Hadamard product. Equation (2) can be rewritten in homogeneous coordinates as:

$$f(\mathbf{X}) = \mathbf{X} \times (\mathbf{P} \circ \boldsymbol{\alpha}) \times \boldsymbol{\beta} \qquad (3)$$

where $\mathbf{X}$ incorporates a column of 1 and we consider an additional group (with index $K+1$) having a single α_{M+1} variable in it. Given the constraints on α variables, α_{M+1} is forced to assume a value equal to 1 and the value of t is then totally incorporated into β_{K+1}. In the following we will assume that the problem is given in homogeneous coordinates and that the constants M and K already count the additional group and variable.

Definition 1. *The partitioned least squared (PartitionedLS) problem is formulated as:*

$$\begin{aligned} \textit{minimize}_{\boldsymbol{\alpha},\boldsymbol{\beta}} & \|\mathbf{X} \times (\mathbf{P} \circ \boldsymbol{\alpha}) \times \boldsymbol{\beta} - \boldsymbol{y}\|_2^2 \\ \text{s.t.} \quad & \boldsymbol{\alpha} \succeq 0 \\ & \mathbf{P}^T \times \boldsymbol{\alpha} = \mathbf{1} \end{aligned}$$

In summary, we want to minimize the squared residuals of $f(\mathbf{X})$, as defined in (3), under the constraint that for each subset k in the partition, the set of weights form a distribution: they need to be all nonnegative as imposed by $\boldsymbol{\alpha} \succeq 0$ constraint and they need to sum to 1 as imposed by $\mathbf{P}^T \times \boldsymbol{\alpha} = \mathbf{1}$ constraint.

Unfortunately we do not know a closed form solution for this problem. Furthermore, the problem is not convex and hence hard to optimally solve using standard out-of-the-box solvers. The following theorem states this fact formally. Due to space constraints we do not provide the proof in full details.

Theorem 1. *The PartitionedLS problem is not convex.*

Proof. (sketch) It suffices to show that the Hessian of the objective function is not positive semidefinite. By Schwarz's theorem, since the loss function has continuous second partial derivatives, the matrix is symmetric and we can apply the Sylvester criterion for checking positive definiteness. In practice, we prove that Hessian is not positive semidefinite by showing that not all leading principal minors are larger than zero. In our specific case, the second minor can be shown

to assume values smaller than zero and this proves the theorem. Let us denote with L the objective of the PartitionedLS problem

$$\begin{aligned} L &= \|\mathbf{X} \times (\mathbf{P} \circ \boldsymbol{\alpha}) \times \boldsymbol{\beta} - \boldsymbol{y}\|_2^2 \\ &= \sum_n \left(\sum_k \beta_k \sum_{\alpha_m \in P_k} \alpha_m x_{n,m} - y_n \right)^2 \end{aligned}$$

Consider the vector containing all the variables of the PartitionedLS problem in the following order: $(\alpha_1, \beta_1, \alpha_2, \beta_2, \ldots, \alpha_K, \beta_K, \alpha_{K+1}, \alpha_{K+2}, \ldots, \alpha_M)$ and assume the problem is not trivial, i.e., that $m > 1, k > 1$. In the following, without loss of generality, we will assume that $\alpha_1 \in P_1$. Under these assumptions, to prove that the second minor is smaller than zero, amounts to prove that:

$$\begin{aligned} H_{11}H_{22} - H_{12}H_{21} &= \frac{\partial^2 L}{\partial \alpha_1 \partial \alpha_1} \frac{\partial^2 L}{\partial \beta_1 \partial \beta_1} - \frac{\partial^2 L}{\partial \alpha_1 \partial \beta_1} \frac{\partial^2 L}{\partial \beta_1 \partial \alpha_1} \\ &= \frac{\partial^2 L}{\partial^2 \alpha_1} \frac{\partial^2 L}{\partial^2 \beta_1} - \left(\frac{\partial^2 L}{\partial \alpha_1 \partial \beta_1} \right)^2 < 0 \end{aligned}$$

By working out the details of the partial derivatives, one ends up with the expression:

$$\begin{aligned} H_{11}H_{22} - H_{12}H_{21} = &\left(2\beta_1^2 \sum_n x_{n,1}^2 \right) 2 \sum_n \left(\sum_{\overline{m} \in P_1} \alpha_{\overline{m}} x_{n,\overline{m}} \right)^2 \\ &- \left[2 \sum_n x_{n,1} \left(\beta_1 \sum_{\overline{m} \in P_1} \alpha_{\overline{m}} x_{n,\overline{m}} + \rho_{\alpha,\beta}(n) \right) \right]^2, \end{aligned} \tag{4}$$

where $\rho_{\alpha,\beta}(n)$ is a short hand for $\sum_k \beta_k \sum_{\overline{m} \in P_k} \alpha_{\overline{m}} x_{n,\overline{m}} - y_n$. To simplify the algebra, let us now assume that for all n, k : $\sum_{\overline{m} \in P_k} \alpha_{\overline{m}} x_{n,\overline{m}}$ is equal to a constant c. We notice that albeit being a strong assumption, it does not hinder the generality of the result since to prove that the Hessian is not semidefinite it suffices to find a single configuration of the problem in which it is not. Under this assumption, $\rho_{\alpha,\beta}(n) = c \sum_k \beta_k - y_n$:

$$\begin{aligned} (4) =& 4Nc^2\beta_1^2 \sum_n x_{n,1}^2 - \left[2 \sum_n x_{n,1} \left(\beta_1 c + c \sum_k \beta_k - y_n \right) \right]^2 \\ =& 4Nc^2\beta_1^2 \sum_n x_{n,1}^2 - \left[2\beta_1 c \sum_n x_{n,1} + 2c \left(\sum_k \beta_k \right) \left(\sum_n x_{n,1} \right) - 2 \sum_n x_{n,1} y_n \right]^2. \end{aligned}$$

We end the proof by noticing that the expression on the left of the minus sign is constant w.r.t. $\beta_2 \ldots \beta_K$, while the part on the right of the minus sign can be made arbitrarily large by varying those variables. This shows that for a certain configuration of β_k values, the expression can be made negative. □

In the following we will provide two algorithms that solve the above problem. One is an alternating least squares approach which scales well with K, but it is not guaranteed to provide the optimal solution. The other one is a reformulation of the problem through a (possibly) large number of convex problems whose minimum is guaranteed to be the optimal solution of the original problem. Even though the second algorithm does not scale well with K, we believe that this *should not be a problem* since the PartitionedLS is by design well suited for small K values (otherwise the main reason inspiring its creation would cease to exist since for large K values the new model would not be much more interpretable than the original one).

3 Algorithms

3.1 Alternating Least Squares approach

In the PartitionedLS problem we aim at minimizing a non convex objective, where the non convexity depends on the multiplicative interaction between α and β variables in the expression $\|\mathbf{X} \times (\mathbf{P} \circ \boldsymbol{\alpha}) \times \boldsymbol{\beta} - \boldsymbol{y}\|_2^2$. Interestingly, if one fixes $\boldsymbol{\alpha}$, the expression $\mathbf{X} \times (\mathbf{P} \circ \boldsymbol{\alpha})$ results in a matrix $\mathbf{X}'$ that does not depend on any variable. Then, the whole expression can be rewritten as a problem $p_{\boldsymbol{\alpha}} = \|\mathbf{X}'\boldsymbol{\beta} - \boldsymbol{y}\|_2^2$ which is the convex objective of a standard least squares problem in the β variables. In a similar way, it can be shown that by fixing $\boldsymbol{\beta}$ one also ends up with a $p_{\boldsymbol{\beta}}$ convex optimization problem.

These observations naturally lead to the formulation of an alternating least squares solution where one alternates between solving $p_{\boldsymbol{\alpha}}$ and $p_{\boldsymbol{\beta}}$. In Algorithm 1 we formalize this intuition into an algorithm where, after initializing $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ randomly, we iterate T times. At each iteration we take the latest estimate for the α variables and solve the $p_{\boldsymbol{\alpha}}$ problem based on that estimate, we then keep the newly found β variables and solve the $p_{\boldsymbol{\beta}}$ problem based on them. At each iteration the overall objective is guaranteed not to increase in value and we conjecture convergence to some stationary point as $T \to \infty$.

3.2 Reformulation as a set of convex subproblems

Here we show how the PartitionedLS problem can be reformulated as a set of convex problems such that the problem of achieving the smallest objective attains the global optimum of the original problem.

Definition 2. *The* PartitionedLS-b problem *is a PartitionedLS problem in which the β variables are substituted by a constant vector $\boldsymbol{b} \in \{-1, 1\}^K$, and the normalization constraints over the α variables are dropped:*

$$\begin{aligned} &\textit{minimize}_{\boldsymbol{\alpha}} \|\mathbf{X} \times (\mathbf{P} \circ \boldsymbol{b}) \times \boldsymbol{\alpha} - \boldsymbol{y}\|_2^2 \\ &\text{s.t.} \quad \boldsymbol{\alpha} \succeq 0 \end{aligned}$$

We note that the above definition actually defines 2^K minimization problems, one for each of the possible $\boldsymbol{b}$ vectors. Interestingly, each one of the

Algorithm 1: Alternating least squares solution to the PartitionedLS problem. The notation $\texttt{const}(\boldsymbol{\alpha})$ (respectively $\texttt{const}(\boldsymbol{\beta})$) is just to emphasize that the current value of $\boldsymbol{\alpha}$ (respectively $\boldsymbol{\beta}$) will be used as a constant in the following step.

```
1   function PartitionedLS-alternating(X, y, P)
2     α = random(M)
3     β = random(K)
4
5     for t in 1...T
6       a = const(α)
7       p* = minimize_β(‖(X × (P ∘ a) × β − y‖₂²))
8
9
10      b = const(β)
11      p* = minimize_α(‖(X × (P ∘ α) × b − y‖₂²,
12                α ⪰ 0,
13                Pᵀ × α = 1)
14    end
15
16    return (p*, α, β)
17  end
```

minimization problems can be shown to be convex by the same argument used in Section 3.1 (for fixed β variables) and we will prove that the minimum attained by minimizing those problems corresponds to the global minimum of the original problem. We also show that by simple algebraic manipulation of the result found by a PartitionedLS-b solution, it is possible to write a corresponding PartitionedLS solution attaining the same objective.

The main breakthrough here derives from noticing that in the original formulation the β variables are used to keep track of two facets of the solution: *i)* the magnitude and *ii)* the sign of the contribution of each subset in the partition of the variables. With the $\boldsymbol{b}$ vector keeping track of the signs, one only needs to reconstruct the magnitude of the β contributions to recover the solution of the original problem.

To do so, let us start by calculating a normalization vector $\bar{\boldsymbol{\beta}}$ containing in $\bar{\beta}_k$ the normalization factor for variables in partition subset k:

$$\bar{\boldsymbol{\beta}} = \left(\sum_{m \in \boldsymbol{P}_k} \alpha_m \right)_k = \mathbf{P}^T \times \boldsymbol{\alpha}.$$

Then, the vector $\hat{\boldsymbol{\alpha}}$ (containing the α variables as defined in the original prob-

Algorithm 2: PartitionedLS-b solution to the PartitionedLS problem. The function `extract_min` retrieves the $(\dot{p}, \dot{\boldsymbol{\alpha}}, \dot{\boldsymbol{\beta}})$ tuple in the results array attaining the lowest $\dot{p}$ value.

```
1   function PartitionedLS-optimal(X, y, P)
2     results = []
3
4     for ḃ in {1, −1}^K
5       ṗ = minimize_α̇(‖(X × (P ∘ α̇) × ḃ − y‖₂²), α̇ ⪰ 0)
6
7       results += (ṗ, α̇, ḃ)
8     end
9
10    p*, α, b = extract_best(results)
11
12
13    β̄ = P^T × α
14    α̂ = (P ∘ α ⊘ β̄^T) × 1
15    β̂ = b ∘ β̄
16
17    return (p*, α̂, β̂)
18  end
```

lem) can be recovered by dividing each α_m by $\bar{\beta}_{k[m]}$:

$$\hat{\boldsymbol{\alpha}} = \left(\frac{\alpha_m}{\bar{\beta}_{k[m]}}\right)_m = \sum_{k=1}^{K} \left((\mathbf{P} \circ \boldsymbol{\alpha}) \oslash \bar{\boldsymbol{\beta}}^T\right)_{\bullet,k} = \left(\mathbf{P} \circ \boldsymbol{\alpha} \oslash \bar{\boldsymbol{\beta}}^T\right) \times \mathbf{1},$$

and the $\hat{\boldsymbol{\beta}}$ vector (containing both signs and magnitudes of the contribution of each subset in the partition) can be reconstructed simply by taking the Hadamard product of $\boldsymbol{b}$ and $\bar{\boldsymbol{\beta}}$:

$$\hat{\boldsymbol{\beta}} = \boldsymbol{b} \circ \bar{\boldsymbol{\beta}}.$$

The complete algorithm, which detects and returns the best solution of the PartitionedLS-b problems over all possible $\boldsymbol{b}$ vectors, is reported in Algorithm 2.

The following lemma (whose proof we omit due to space constraints) shows that a PartitionedLS solution using $\hat{\boldsymbol{\alpha}}$ and $\hat{\boldsymbol{\beta}}$ has the same objective value as the PartitionedLS-b solution using the given $\boldsymbol{b}$ and $\boldsymbol{\alpha}$ values.

Lemma 1. *(Rewriting Lemma) Let $\boldsymbol{\alpha}$ be a vector of m positive values, $\boldsymbol{b} \in \{-1, 1\}^K$ a vector of K signs, and $\bar{\boldsymbol{\beta}}$ a vector of K non zero values. Let also $\hat{\boldsymbol{\alpha}}$,*

$\hat{\boldsymbol{\beta}}$ *be such that:*

$$\hat{\boldsymbol{\alpha}} = \left(\frac{\alpha_m}{\bar{\beta}_{k[m]}} \right)_m \quad \textit{for } m \in \{1 \dots M\}$$

and

$$\hat{\boldsymbol{\beta}} = \left(b_k \bar{\beta}_k \right)_k \ \textit{for } k \in \{1 \dots K\}.$$

Then:

$$\mathbf{X} \times (\mathbf{P} \circ \boldsymbol{\alpha}) \times \boldsymbol{b} = \mathbf{X} \times (\mathbf{P} \circ \hat{\boldsymbol{\alpha}}) \times \hat{\boldsymbol{\beta}}.$$

Corollary 1. *Under the hypotheses of the Rewriting Lemma it holds:*

$$\|\mathbf{X} \times (\mathbf{P} \circ \hat{\boldsymbol{\alpha}}) \times \hat{\boldsymbol{\beta}} - \boldsymbol{y}\|_2^2 = \|\mathbf{X} \times (\mathbf{P} \circ \boldsymbol{\alpha}) \times \boldsymbol{b} - \boldsymbol{y}\|_2^2 \tag{5}$$

Theorem 2. *Let p^* be the optimal value of the PartitionedLS problem and let p_{b^*} be the value attained by the PartitionedLS-b algorithm (Algorithm 2). Then, $p^* = p_{b^*}$.*

Proof. We first show that $p^* \geq p_{b^*}$, then we show that $p_{b^*} \leq p^*$ and conclude that $p^* = p_{b^*}$. In the following let:

- $\boldsymbol{b}^*$ be the best sign vector as found by Algorithm 2 and let $\boldsymbol{\alpha}_{b^*}$ be the corresponding $\boldsymbol{\alpha}$ vector (i.e., $\boldsymbol{\alpha}_{b^*}, \boldsymbol{b}^*$ attain the p_{b^*} solution);
- $\hat{\boldsymbol{\alpha}}^*, \hat{\boldsymbol{\beta}}^*$ be the values attaining the p^* solution.

Notice that Corollary 1 of the Rewriting Lemma implies that for sign vector $\boldsymbol{b} = \hat{\boldsymbol{\beta}} \oslash \bar{\boldsymbol{\beta}}$ and $\boldsymbol{\alpha}_b = \left(\alpha_m \bar{\beta}_{k[m]} \right)_m$:

$$p^* = \|\mathbf{X} \times (\mathbf{P} \circ \hat{\boldsymbol{\alpha}}^*) \times \hat{\boldsymbol{\beta}}^* - \boldsymbol{y}\|_2^2 = \|\mathbf{X} \times (\mathbf{P} \circ \boldsymbol{\alpha}_b) \times \boldsymbol{b} - \boldsymbol{y}\|_2^2.$$

Since the p_{b^*} solution is the best solution over all the possible sign vectors, it holds that:

$$\|\mathbf{X} \times (\mathbf{P} \circ \boldsymbol{\alpha}_b) \times \boldsymbol{b} - \boldsymbol{y}\|_2^2 \geq \|\mathbf{X} \times (\mathbf{P} \circ \boldsymbol{\alpha}_{b^*}) \times \boldsymbol{b}^* - \boldsymbol{y}\|_2^2 = p_{b^*}.$$

Vice-versa by Corollary 1 of the Rewriting Lemma it holds that for $\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\beta}}$ as given in the Rewriting Lemma assumptions, it holds that:

$$p_{b^*} = \|\mathbf{X} \times (\mathbf{P} \circ \boldsymbol{\alpha}_{b^*}) \times \boldsymbol{b}^* - \boldsymbol{y}\|_2^2 = \|\mathbf{X} \times (\mathbf{P} \circ \hat{\boldsymbol{\alpha}}) \times \hat{\boldsymbol{\beta}} - \boldsymbol{y}\|_2^2.$$

Since p^* is the global optimum for the PartitionedLS problem, it holds:

$$\|\mathbf{X} \times (\mathbf{P} \circ \hat{\boldsymbol{\alpha}}) \times \hat{\boldsymbol{\beta}} - \boldsymbol{y}\|_2^2 \geq \|\mathbf{X} \times (\mathbf{P} \circ \hat{\boldsymbol{\alpha}}^*) \times \hat{\boldsymbol{\beta}}^* - \boldsymbol{y}\|_2^2 = p^*$$

□

4 Regularization

The PartitionedLS model presented so far has no regularization mechanism in place and, as such, it risks overfitting the training set. Since the α values are normalized by definition, the only parameters that need regularization are those collected in the $\boldsymbol{\beta}$ vector. Then, the regularized version of the objective function simply adds a penalty on the size of the $\boldsymbol{\beta}$ vector:

$$\|\mathbf{X} \times (\mathbf{P} \circ \boldsymbol{\alpha}) \times \boldsymbol{\beta}\|_2^2 + \eta \|\boldsymbol{\beta}\|_2^2 \tag{6}$$

where the squared euclidean norm could be substituted with the L1 norm in case a LASSO-like regularization is preferred.

The objective expressed in (6) can be used in Algorithm 1 as is, but it needs to be slightly updated so to accommodate the differences in the objective function when used in Algorithm 2. In this second case, in fact, the correct expression for the $\|\boldsymbol{\beta}\|_2^2$ regularization term becomes: $\|\mathbf{P}^T \times \boldsymbol{\alpha}\|_2^2$ since the optimization program does not maintain an explicit list of β variables. We notice that since the regularization term is convex, it does not hinder the convexity of the optimization problems in both algorithms presented in this paper.

5 Experiments

While the main motivation of the proposed approach is interpretability, we do not provide here any direct measurement of this property. Unfortunately, interpretability is not easily measurable since its very notion has not yet been clearly defined and a multitude of different definitions coexist. Instead, we simply argue that the smaller "grouped" model better matches one interpretability definition based on *transparency* (in both the *simulatability* and *decomposability* meanings, see [7]). In the following we will focus on the algorithmic properties of the two algorithms we presented in this paper, showing how they behave so to provide some insight about when one should be preferred over the other.

In order to assess the advantages/disadvantages of the two algorithms presented in this paper, we apply them to solve the block-relevance analysis proposed in [4, 3]. We will assess the two algorithms on a dataset [2] containing 82 features describing measurements over simulated (VolSurf+ [5]) models of 44 drugs. The regression task is the prediction of the lipophilicity of the 44 compounds. The 82 features are partitioned into 6 groups according to the kind of property they describe. The six groups are characterized in [3] as follows:

- **Size/Shape**: 7 features describing the size and shape of the solute;
- **OH2**: 19 features expressing the solute's interaction with water molecules;
- **N1**: 5 features describing the solute's ability to form hydrogen bond interactions with the donor group of the probe;
- **O**: 5 features expressing the solute's ability to form hydrogen bond interactions with the acceptor group of the probe;

- **DRY**: 28 features describing the solute's propensity to participate in hydrophobic interactions;
- **Others**: 18 descriptors describing mainly the imbalance between hydrophilic and hydrophobic regions.

This dataset, while not high-dimensional in the broadest sense of the term, can be partitioned into well-defined, interpretable groups of variables. Previous literature which employed this dataset has indeed focused on leveraging the data's structure to obtain explainable results [4].

We used as training/test split the one proposed in [2] and set the regularization parameter η to 1.0 (since we are not aiming at finding the most accurate regressor, we did not investigate other regularization settings).

For this particular problem, the number of groups is small and the optimal algorithm needs to solve just $2^6 = 64$ convex problems. It terminates in 0.90 seconds reaching a value of the regularized objective function of about 6.679. For what concerns the approximated algorithm, we ran it in a Multistart fashion with 100 randomly generated starting points. We repeated the experiment using two different configurations of parameter T (number of iterations), setting it to 20 and 100, respectively. For each configuration we kept track of the cumulative time and of the ("cumulative") best solution found. As one would expect, increasing the value of parameter T slows down the algorithm, but allows it to converge to better solutions. Figure 1 reports the best objective value found by the algorithms plotted against the time (reported on a logarithmic scale to improve visualization) necessary to get to such a solution. The experiments show that Algorithm 2 retrieves a more accurate (actually the globally optimal) answer in a fraction of the time. Indeed, it is straightforward to observe that, in typical scenarios[1], the only times where the alternating least squares approach outperforms the optimal algorithm in terms of running time is for cases where the total number of iterations (convex subproblems solved) is smaller than the 2^K subproblems needed by Algorithm 2 to compute the optimal solution. In our admittedly limited experimentation, this leads to solutions that grossly approximate the optimal one. Our conclusion is that the optimal algorithm is likely to be preferable in most cases. The exceptions are the cases where the number of groups is large or the cases where the time required to solve a single convex problem is very large and approximate solutions do not hinder the applicability of the result in the application at hand. For what concerns the cases with a large number of groups, we argue that this setting defies the main motivation behind employing a model such as the one we presented.

[1] In this informal argument we are assuming that each convex problem requires about the same amount of time to be solved. While this is not guaranteed, we believe that it is very unlikely that deviations from this assumption would lead to situations very different from the ones outlined in the argument.

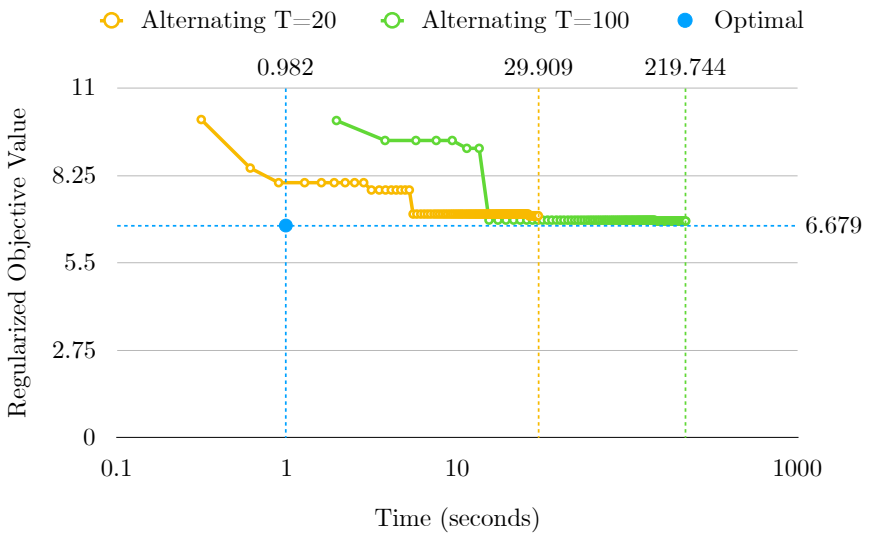

Figure 1: Plot of the behavior of the two proposed algorithms. The PartitionedLS-alternating algorithm has been repeated 100 times following a Multistart strategy and in two settings (T=20 and T=100). Each point on the orange and green lines reports the cumulative time and best objective found during these 100 restarts.

6 Conclusions

In this paper, we presented an alternative least squares linear regression formulation and two algorithms to solve it: one iterative, one optimal. In our experimentation, we found the optimal algorithm to be faster, although its time complexity grows exponentially with the number of groups. Our model enables scientists and practitioners to group features together into partitions modeling higher level abstractions which are easier to reason about. In the future, we would like to perform an extensive experimentation on high-dimensionality data to better understand the tradeoff between our Alternating Least Squares and exact approaches when the number of groups is higher. In contrast with the choice we made in this paper (where we focused on a dataset showcasing a real application of the methodology), in this new experimentation we will use more broadly available datasets. Even though the datasets and groupings will appear to be less justified, the new setting will allow us to better study in which cases it may be more beneficial to recover a lower-quality solution in a shorter amount of time, rather than striving for an optimal solution using the exact algorithm (Algorithm 2). We also plan to investigate branch-and-bound strategies to avoid the explicit computation of all 2^K sub-problems in the optimal algorithm.

A Julia [1] implementation of the algorithms is available at `https://github.com/ml-unito/PartitionedLS`; the code for the experiments can be downloaded from: `https://github.com/ml-unito/PartitionedLS-experiments`.

This latter repository also contains the dataset we used in our experiments in the format required to be loaded from the programs. Due to technical reasons, the original dataset presented in [2] is no longer available for download. The authors confirmed that they are willing to provide the data to interested researchers if contacted directly.

References

[1] Bezanson, J., Karpinski, S., Shah, V.B., Edelman, A.: Julia: A fast dynamic language for technical computing. CoRR **abs/1209.5145** (2012), http://arxiv.org/abs/1209.5145

[2] Caron, G., Vallaro, M., Ermondi, G., Goetz, G.H., Abramov, Y.A., Philippe, L., Shalaeva, M.: A fast chromatographic method for estimating lipophilicity and ionization in nonpolar membrane-like environment. Molecular Pharmaceutics **13**(3), 1100–1110 (2016). https://doi.org/10.1021/acs.molpharmaceut.5b00910, https://doi.org/10.1021/acs.molpharmaceut.5b00910, pMID: 26767433

[3] Ermondi, G., Caron, G.: Molecular interaction fields based descriptors to interpret and compare chromatographic indexes. Journal of Chromatography A **1252**, 84 – 89 (2012). https://doi.org/https://doi.org/10.1016/j.chroma.2012.06.069, http://www.sciencedirect.com/science/article/pii/S0021967312009636

[4] Giulia, C., Maura, V., Giuseppe, E.: The block relevance (br) analysis to aid medicinal chemists to determine and interpret lipophilicity. Med. Chem. Commun. **4**, 1376–1381 (2013). https://doi.org/10.1039/C3MD00140G

[5] Goodford, P.J.: A computational procedure for determining energetically favorable binding sites on biologically important macromolecules. Journal of Medicinal Chemistry **28**(7), 849–857 (1985). https://doi.org/10.1021/jm00145a002, https://doi.org/10.1021/jm00145a002

[6] Huang, J., Breheny, P., Ma, S.: A selective review of group selection in high-dimensional models. Statistical science: a review journal of the Institute of Mathematical Statistics **27**(4) (2012)

[7] Lipton, Z.: The mythos of model interpretability. Communications of the ACM **61** (10 2016). https://doi.org/10.1145/3233231

[8] Wold, S., Sjöström, M., Eriksson, L.: Pls-regression: a basic tool of chemometrics. Chemometrics and Intelligent Laboratory Systems **58**(2), 109 – 130 (2001). https://doi.org/https://doi.org/10.1016/S0169-7439(01)00155-1, http://www.sciencedirect.com/science/article/pii/S0169743901001551, pLS Methods