Closed Form Solution for Quadratic Programs with Equality Constraints

Pietro Calabrese[a], Francesco Marabotto[a], Lorenzo Fiaschi[a,*] and Marco Cococcioni[a]

[a] *Department of Information Engineering, Largo Lucio Lazzarino 1, Pisa, 56122, Italy*

ARTICLE INFO

Keywords:
Quadratic Programming
Non-Standard Analysis
Non-Archimedean Scientific Computing
Alpha Theory

ABSTRACT

In this letter we provide a closed formula for the solution of any least squares problem with equality constraints. To this end, we leverage a non-Archimedean model, therefore on a set of numbers that contain not only finite numbers, but also infinitely large and infinitely small numbers. The theoretical results are accompanied by empirical verifications. In particular we will focus on Least Squares problems.

1. Introduction

We are interested in problems with this form:

$$\begin{aligned} \min_{w} \quad & \frac{1}{2} w^T Q w + c^T w \\ \text{s.t.} \quad & Aw = b, \end{aligned}$$

where $Q = X^T X$ is a positive definite matrix, $X \in \mathbb{R}^{m \times n}$, $c = X^T y$ is a real vector, $y \in \mathbb{R}^m$, $A \in \mathbb{R}^{p \times n}$ and $b \in \mathbb{R}^p$. In the literature ([4], [5], [6], [7]) there are several ideas and approaches to find a closed form solution, so a way to find the optimal solution $\bar{w}$ of this problem using a certain combination of $f(A, b, Q, c)$. However, for several reasons, there is not a unique version of this solution. In the literature there are proposals which however require either hypotheses on the form of the problem or which are parametric. In this paper we propose a solution in closed form, independent of the form of the problem and without parameters.

We will start, in Section 2, from the presentation of three approaches present in the literature and which find a solution to optimization problems with equality constraints, but this solution is not in a closed form. In Section 3 we introduce BAN numbers and some notions of non-standard mathematics, necessary for the development and implementation of the work. In Section 4 we will present the solution in a closed form. Sections 5 introduces the possibility to update the current solution by adding (removing) observations and constraints to (from) the problem. In Section 5.2 we will present a way of calculating the update of the matrix M in an efficient way. Section 6 presents the results we obtained in the empirical investigations on synthetic generated data starting from a suitably chosen optimal solution. This way of proceeding has been chosen (i.e. generating the data starting from the optimal solution) in such a way as to be able to easily demonstrate that the equation of the curve provided by the solution in closed form corresponds to the optimal solution appropriately chosen at the beginning. Section 7 concludes the work by providing a summary of what has been done.

*Corresponding author

p.calabrese2@studenti.unipi.it (P. Calabrese); f.marabotto@studenti.unipi.it (F. Marabotto); lorenzo.fiaschi@phd.unipi.it (L. Fiaschi); m.cococcioni@unipi.it (M. Cococcioni)

ORCID(s):

2. Solving linear least squares problems with linear equality constraints

In this section we present the three main approaches that have been developed in the literature as closed-form solutions of Least Squares problems. In particular, we focus our attention on Least Squares problems as they present a more interesting application; indeed, there is a need to add and remove data in order to modify the temporary solution. The approaches we report in this section are presented in detail in [7] and in chapters 20, 21 and 22 of the book [6]. We assume that $X \in \mathbb{R}^{m \times n}$ and $A \in \mathbb{R}^{p \times n}$, with $m > n \gg p$. Given $y \in \mathbb{R}^m$ and $b \in \mathbb{R}^p$, the least squares problem with equality constraints (the LSE problem) is

$$\min_{w \in \mathbb{R}^n} \quad \|Xw - y\|_2^2 \tag{2.1a}$$

$$\text{s.t.} \quad Aw = b, \tag{2.1b}$$

2.1. The null-space approach

The null-space approach is a standard technique for solving 2.1a. It is based on constructing a matrix $Z \in \mathbb{R}^{n \times (n-p)}$ such that its columns form a basis for $\mathcal{N}(A)$, that denotes the null space of matrix A. Any $w \in \mathbb{R}^n$ satisfying the constraints can be written in the form

$$w = w_1 + Z w_2, \tag{2.2}$$

where $w_1 \in \mathbb{R}^n$ is a particular solution of the underdetermined system $Aw_1 = b$. The minimum norm solution can be obtained from the QR factorization of A, that is, $AP = Q(R_A 0)$, where the permutation $P \in \mathbb{R}^{x \times n}$ represents the pivoting, $R \in \mathbb{R}^{p \times p}$ is an upper triangular matrix and $Q \in \mathbb{R}^{p \times p}$ is an orthogonal matrix. w_1 is given by

$$w_1 = P \begin{pmatrix} R^{-1} Q^T b \\ 0 \end{pmatrix}$$

Substituting 2.2 into 2.1a gives the transformed LS problem

$$\min_{w_2} \quad ||XZw_2 - (y - Xw_1)||^2$$

where the closed form solution is

$$\bar{w} = A^T b + (XZ)^T (y - XA^T b).$$

This solution is not a real closed form since to calculate it not only the input data are used, but it is necessary to calculate the QR factorization of matrix A and to find the permutation matrix for A.

2.2. The method of direct elimination

The basic idea is to express the dependency of p selected components of the vector w on the remaining $n-p$ components and to substitute this into the LS problem 2.1a. Here it is proposed how to choose the p components so as to retain sparsity in the transformed problem. Consider the Constraints 2.1b. The method starts by permuting and splitting the solution components as follows:

$$Aw = AP_A\tau = \begin{pmatrix} A_1 & A_2 \end{pmatrix} \begin{pmatrix} \tau_1 \\ \tau_2 \end{pmatrix} = b,$$

where $P_A \in \mathbb{R}^{n\times n}$ is a permutation matrix chosen so that $A_1 \in \mathbb{R}^{p\times p}$ is nonsingular. Let $XP_A = \begin{pmatrix} X_1 & X_2 \end{pmatrix}$ be a conformal partitioning of XP_A. Substituting the expression

$$\tau_1 = A_1^{-1}(b - A_2\tau_2)$$

into 2.1a gives the transformed LS problem

$$\min_{\tau_2} \quad \|X_T\tau_2 - (y - X_1A_1^{-1}b)\|_2^2$$

with the transformed matrix

$$X_T = X_2 - X_1A_1^{-1}A_2 \in \mathbb{R}^{m\times(n-p)}.$$

The closed form solution is

$$\tau_1 = A_1^{-1}(b - A_2\tau_2)$$

Also this solution does not appear in a real closed form since it requires the inspection of the matrix A until an invertible minor is found.

2.3. Approaches described via augmented systems

We now focus on complementary approaches that are based on substitution from the unconstrained least squares problem into the constraints. A useful way to describe this is via the augmented (or saddle-point) system:

$$\begin{pmatrix} H & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} w \\ \lambda \end{pmatrix} = \begin{pmatrix} X^T y \\ b \end{pmatrix}, \qquad H = X^T X \quad (2.3)$$

Here $\lambda \in \mathbb{R}^p$ is a vector of additional variables that are often called *Lagrange multipliers*.

$$\begin{bmatrix} A \\ \epsilon X \end{bmatrix} w \cong \begin{bmatrix} b \\ \epsilon y \end{bmatrix}$$

solves the 2.3 LSE problem, with $0 < \epsilon \ll 1$.

With this approach there is no guarantee of finding the optimal solution; furthermore, it may suffer from instability and may require appropriate row swapping.

3. Alpha Theory: an introduction

Our approach is based on Alpha-Theory (AT). Alpha Theory is a non-standard model of numbers, that is, a theory that allows the use of numerical sets larger than $\mathbb{R}$, that is, sets that contain not only $\mathbb{R}$ but also infinitely large and infinitely small numbers. Hence also the non-Archimedean appellation, since Archimedes' axiom is not satisfied. In this section we will summarize some fundamental concepts of Aplha Theory, necessary to make the discussion understandable. A reference to Alpha Theory is [1]. The key concept is that Alpha Theory fixes a point at infinity α and constructs all the numbers in this field larger than $\mathbb{R}$ and which we call $\mathbb{E}$ as a kind of alpha polynomial. As happens for the real ones that are represented in a machine with an encoding called IEEE 754, also for these numbers it is possible to realize a numerical encoding, with finite dimension, which allows to perform operations on a computer; this encoding is called BAN (Bounded Algorithmic Number).

Definition 1. *(Algoritimic number) A number $\xi \in \mathbb{E}$ is called algorithmic if it can be represented as a finite sum of monosemia, namely,*

$$\xi = \sum_{k=0}^{l} r_k\alpha^{s_k}; r_k \in \mathbb{R}, s_k \in \mathbb{Q}; s_k > s_{k+1}. \qquad (3.1)$$

Moreover, one can always represent it in the following form, called "normal form":

$$\xi = \alpha^p P\left(\eta^{\frac{1}{m}}\right)$$

where

$$\eta := \alpha^{-1},$$

$p \in \mathbb{Q}$, $m \in \mathbb{N}$ and $P(x)$ is a polynomial with real coefficients such that $P(0) \neq 0$.

Notice that this is not a symbolic approach, but a numerical one, as the encoding is of a fixed dimension, so it can obtain performances, in terms of time and computational resources, superior to those obtained through symbolic tools such as *Wolfram Mathematica*. Work [3] shows more details about it.

4. The Closed Form Solution

In this section we introduce the motivation that led to the use of BANs to find a closed-form solution to quadratic problems with equality constraints. Suppose we have the following standard optimization problem:

$$\begin{aligned} \min \quad & \|Xw - y\|_2^2 \\ \text{s.t.} \quad & Aw = b, \end{aligned} \qquad (4.1)$$

We can write any regression problem as

$$\begin{aligned} \min_{w} \quad & \|w\|_2^2 \\ \text{s.t.} \quad & Xw = y \end{aligned} \qquad (4.2)$$

Solve the following problem

$$\min \quad \|Xw - y\|_2^2 + \|w\|_2^2 \tag{4.3}$$

is a typical approach to solve 4.2. Problem 4.3 admits the closed form solution reported in the next equation:

$$\bar{w} = (X^T X + I)^{-1} X^T y \tag{4.4}$$

Notice that $\bar{w}$ in 4.4 is just an approximation of the solution of 4.2, since it is the solution of that problem by adding the constraint in the objective function, and the closeness of the two is typically tuned by an hyperparameter $\lambda \in \mathbb{R}^+$ as follows

$$\bar{w} = (X^T X + \lambda I)^{-1} X^T y$$

We added the hyperparameter λ in order to prevent numerical instability. A choice of λ which enforces identity between such solutions is $\lambda = \eta$, i.e., a non-Archimedean one. Thus,

$$\bar{w} = (X^T X + \eta I)^{-1} X^T y$$

coincides with the solution of 4.2, putting η equal to an infinitesimal number, when exists, and is the unique optimal point of the problem

$$\min \quad \|Xw - y\|_2^2 + \eta\|w\|_2^2 \tag{4.5}$$

which is a non-Archimedean least squares problem.

Consider now the case where further constraints are given on w, i.e., $Aw = b$. Such constraints represent “hard” requirements over w and are application-dependent. On the contrary, the requirement $Xw = y$ is supposed to be “soft”, in the sense that if no w can exactly fit the data the choice must fall on the one which does most. Thus, the natural way to enforce this behavior is representing the problem as the following constrained version of 4.3:

$$\begin{aligned} \min \quad & \|Xw - y\|_2^2 + \lambda\|w\|_2^2 \\ \text{s.t.} \quad & Aw = b, \end{aligned} \tag{4.6}$$

To solve such a program, iterative algorithms such as interior point method are typically used. In line with the idea underlying 4.5 however, a non-Archimedean alternative approach is possible, which is non-iterative one: it will give us a closed formula solution to the same problem. It consists in introducing also the feasibility of the solution in the cost function, and its priority over the quality of the regression is enforced by weighting it by α:

$$\min \quad \alpha\|(Aw - b)\|_2^2 + \|Xw - y\|_2^2 + \eta^2\|w\|_2^2$$

which, scaled down by α, assumes the following final form

$$\min \quad \|Aw - b\|_2^2 + \eta\|Xw - y\|_2^2 + \eta^2\|w\|_2^2 \tag{4.7}$$

One of the most interesting properties of 4.7 is that it always admits one unique solution, regardless the feasibility and the degeneracy of the constrained regression, that is

$$\bar{w} = (A^T A + \eta X^T X + \eta^2 I)^{-1}(A^T b + \eta X^T y) \tag{4.8}$$

In order to verify whether or not the original problem is feasible is enough to verify $A\bar{w} = b$. In case the equality holds true, $\bar{w}$ is the optimal point, otherwise the problem is infeasible.

For small matrices, the solution in 4.8 might be evaluated symbolically in environments like *Wolfram Mathematica*, after introducing the symbol η, even if *Mathematica* being aware it is an infinitesimal value. However, when the involved matrices and vectors are large, computing the final solution, which is a standard one, soon becomes impossible ([3]). On the contrary, by using BANs, we are able to compute it, for dimensions higher than the ones that can be handled by *Mathematica*, since BAN uses a finite-size encoding.

5. Online Update

5.1. Incremental Learning

Another very useful feature of the reformulation in 4.7 is that any temporary solution 4.8 can be incrementally updated in case additional observations are available or further constraints are added to the problem. This is achieved preserving the properties of existence and uniqueness of the solution, along with the possibility to easily check the maintenance of problem feasibility.

Theorem 1. *Let w^k be the solution of 4.7 in the form of 4.8, and $(\tilde{x}, \tilde{y})$ be a new observation of the regression problem. Then, the optimal solution w^k+1 of the program considering also the additional observation can be written as $w^k + 1 = w^k + \delta^k$, where*

$$\delta^k = \eta[A^T A + \eta(X^T X + x\tilde{x}^T) + \eta^2 I]^{-1}(\tilde{x}\tilde{y} - \tilde{x}\tilde{x}^T w^k) \tag{5.1}$$

Proof. Proof according to 4.8

$$[A^T A + \eta(X^T X + x\tilde{x}^T) + \eta^2 I]w^{k+1} = ATb + \eta(X^T y + \tilde{x}\tilde{y}),$$
$$(A^T A + \eta X^T X + \eta^2 I)w^k = ATb + \eta X^T y.$$

If one represents $w^{k+1} = w^k + \delta^k$, then it is possible to retrieve δ^k. The following algebraic manipulations give the thesis

$$[A^T A + \eta(X^T X + x\tilde{x}^T) + \eta^2 I](w^k + \delta^k) = A^T b + \eta(X^T y + \tilde{x}\tilde{y})$$
$$\Downarrow$$
$$\eta\tilde{x}\tilde{x}^T w^k + [A^T A + \eta(X^T X + x\tilde{x}^T) + \eta^2 I]\delta^k = \eta\tilde{x}\tilde{y}$$
$$\Downarrow$$
$$\delta^k = \eta[A^T A + \eta(X^T X + x\tilde{x}^T) + \eta^2 I]^{-1}(\tilde{x}\tilde{y} - \tilde{x}\tilde{x}^T w^k).$$

Theorem 2. *Let w^k be the solution of 4.7 in the form of 4.8, and $(\tilde{a}, \tilde{b})$ be a new constraint to the regression problem. Then, the optimal solution w^{k+1} of the program considering also the additional constraint can be written as $w^{k+1} = w^k + \delta^k$, where*

$$\delta^k = \eta(A^T A + \tilde{a}\tilde{a}^T \eta X^T X + \eta^2 I)^{-1}(\tilde{a}\tilde{b} - \tilde{a}\tilde{a}^T w^k) \tag{5.2}$$

PROOF. Proof according to 4.8

$$(A^T A + \tilde{a}\tilde{a}^T + \eta X^T X + \eta^2 I)w^{k+1} = A^T b + X^T y + \tilde{a}\tilde{b},$$

$$(A^T A + \eta X^T X + \eta^2 I)w^k = ATb + \eta X^T y.$$

If one represents $w^{k+1} = w^k + \delta^k$, then it is possible to retrieve δ^k. The following algebraic manipulations give the thesis

$$[A^T A + \tilde{a}\tilde{a}^T + \eta X^T X + \eta^2 I](w^k + \delta^k) = A^T b + \tilde{a}\tilde{b} + \eta X^T y$$

$$\Downarrow$$

$$\tilde{a}\tilde{a}^T w^k + [A^T A + \tilde{a}\tilde{a}^T + \eta X^T X + \eta^2 I]\delta^k = \tilde{a}\tilde{b}$$

$$\Downarrow$$

$$\delta^k = \eta(A^T A + \tilde{a}\tilde{a}^T \eta X^T X + \eta^2 I)^{-1}(\tilde{a}\tilde{b} - \tilde{a}\tilde{a}^T w^k).$$

5.2. Efficient updates

Notice that the inversion of matrices in 5.1 and 5.2 can be efficiently computed leveraging the Sherman-Morrison formula. Indeed, in both the cases it results in the inversion of a matrix of the form $M + uu^T$, where M^{-1} is already known from the previous step and

$$M = A^T A + \eta X^T X + \eta^2 I.$$

According to the Sherman-Morrison formula, it holds true that

$$(M + uu^T)^-1 = M^{-1} - \frac{M^{-1}uu^T M^{-1}}{1 + u^T M^{-1} u}$$

Thus, 5.1 becomes

$$\delta^k = \eta \left[M^{-1} - \frac{M^{-1}\tilde{x}\tilde{x}^T M^{-1}}{1 + \tilde{x}^T M^{-1}\tilde{x}} \right] (\tilde{x}\tilde{y} - \tilde{x}\tilde{x}^T w^k), \quad (5.3)$$

while 5.2 becomes

$$\delta^k = \left(M^{-1} - \frac{M^{-1}\tilde{a}\tilde{a}^T M^{-1}}{1 + \tilde{a}^T M^{-1}\tilde{a}} \right) (\tilde{a}\tilde{b} - \tilde{a}\tilde{a}^T w^k). \quad (5.4)$$

5.3. Incremental Forgetting

As well as in Section 5.1, it may happen that some constraints or data points need to be forgotten. The reason can be multivariate, e.g., environmental conditions change and some new constraints appear as other stop to hold, while there might be the need to forget some observations since they have been recognized as outliers (notice that this last fact might be crucial in applications since $L2$ norm is not robust to outliers). Again, an incremental update of the current solution so as to forget previous information is possible.

Theorem 3. *Let w^k be the solution of 4.7 in the form of 4.8, and $(\tilde{x}, \tilde{y})$ be an observation of the regression problem to forget. For the sake of a clearer notation, assume the data matrix has the form $[\tilde{x}X^T]^T$, and the regression vector has the form $[\tilde{y}y^T]^T$. Otherwise, permute the observations so they meet such an assumption. Then, the optimal solution w^{k+1} of the program once forgotten the observation can be written as $w^{k+1} = w^k - \gamma^k$, where*

$$\gamma^k = \eta(A^T A + \eta X^T X + \eta^2 I)^{-1}(\tilde{x}\tilde{y} - \tilde{x}\tilde{x}^T w^k). \quad (5.5)$$

The corresponding efficient update is

$$\gamma^k = \eta \left(M^{-1} + \frac{M^{-1}\tilde{x}\tilde{x}^T M^{-1}}{\alpha - \tilde{x}^T M^{-1}\tilde{x}} \right) (\tilde{x}\tilde{y} - \tilde{x}\tilde{x}^T w^k). \quad (5.6)$$

where

$$M = A^T A + \eta(X^T X + \tilde{x}\tilde{x}^T) + \eta^2 I$$

which is already known from previous step.

PROOF. The proof is analogous to the one in Theorem 1, and it is omitted for brevity.

Theorem 4. *Let wk be the solution of 4.7 in the form of 4.8, and $(\tilde{a}, \tilde{b})$ be a constraint of the regression problem to forget. For the sake of a clearer notation, assume the constraint matrix has the form $[\tilde{a}A^T]^T$, and the constant term vector has the form $[\tilde{b}b^T]^T$. Otherwise, permute the constraints so they meet such an assumption. Then, the optimal solution wk+1 of the program once forgotten the constraint can be written as $w^{k+1} = w^k - \gamma^k$, where*

$$\gamma^k = (A^T A + \eta X^T X + \eta^2 I)^{-1}(\tilde{a}\tilde{b} - \tilde{a}\tilde{a}^T w^k). \quad (5.7)$$

The corresponding efficient update is

$$\gamma^k = \left(M^{-1} + \frac{M^{-1}\tilde{a}\tilde{a}^T M^{-1}}{1 - \tilde{a}^T M^{-1}\tilde{a}} \right) (\tilde{a}\tilde{b} - \tilde{a}\tilde{a}^T w^k). \quad (5.8)$$

where

$$M = A^T A + \tilde{a}\tilde{a}^T + \eta X^T X + \eta^2 I,$$

which is already known from previous step.

PROOF. The proof is analogous to the one in Theorem 2, and it is omitted for brevity.

6. Experiments

To verify the correctness of what is expressed in Section 4 and Section 5, we carried out tests on numbers synthetically generated. This section will report the results obtained for each of the previous sections. The tests were performed on *Jupyter Notebooks* using *Julia* as a programming language and having included the BAN number library, necessary to correctly implement the mathematical treatment.

6.1. Results about Closed Form Solution

We performed some experiments regarding the closed form solution by proceeding as follows. We fixed the equation of the curve and we generated data adding some random noise to prevent the data from being exactly in the curve, in order to evaluate the quality of the results. Later, thanks to

Julia's Interact package, it was possible to insert two widgets that compose a simple graphical interface that allowed us to modify, in real-time, the number of observations and the number of constraints. In particular, we generated 100 observations and 4 constraints, to see how the closed-form solution adapted to changes in the data of the initial problem. Figures 1 and 2 show the results of two tests that have been carried out.

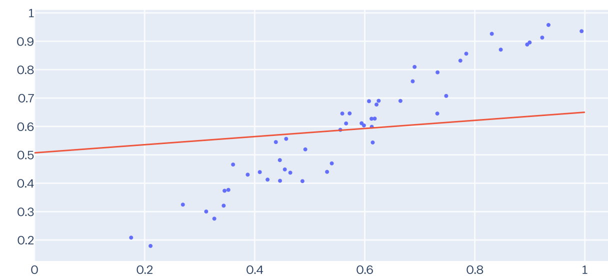


Figure 1: 50 observations, 3^{rd} constraint, equation: $0.143x + 0.507$

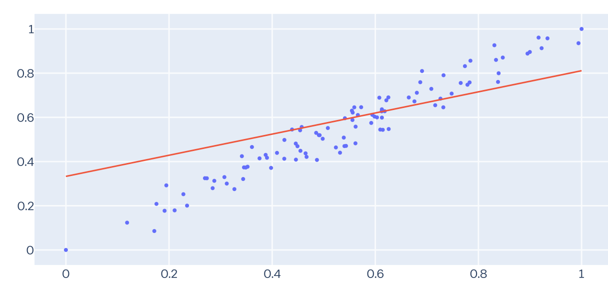


Figure 2: 100 observation, 2^{nd} constraint, equation: $0.478x + 0.333$

Two observations that need to be made are the following. Notice that if the number of observations changes from 50 to 100, there is a recalculation of the problem is from the beginning as if the initial number of observations were 100. Also notice that in this part the number of constraints is always the same (only one constraint). As for adding and removing constraints, these will be presented in Section 5. Before illustrating this section, the results of the quadratic case are shown in Figure 3 and 4. The experiments were carried out in a similar way to those of the previous linear case.

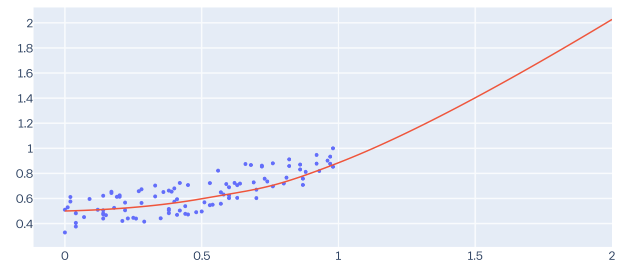


Figure 3: 100 observations, 2^{nd} constraint, equation: $0.379x^2 + 0.056x + 0.5$

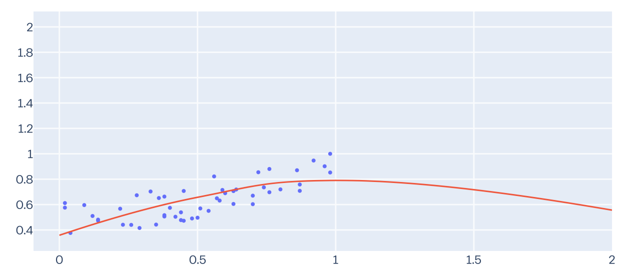


Figure 4: 50 observation, 4^{th} constraint, equation: $-0.333x^2 + 0.765x + 0.358$

6.2. Results about Incremental Learning

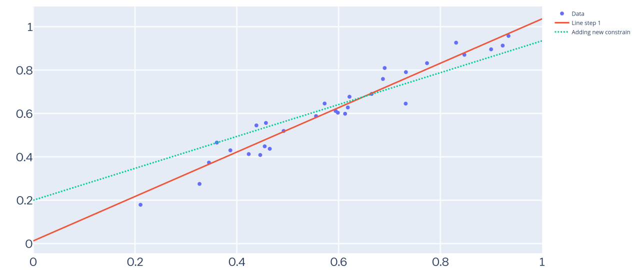
As regards Incremental Learning, we present the results about linear and quadratic case. In Figure 5 there are four images showing the following experiments: starting from the initial situation of the linear case, we added a constraint (image 5a) and three observations (image 5b); starting from the initial situation of the quadratic case, we added two constraints (image 5c) and three observations (image 5d). These images have been reported to show how the red curve evolves with the addition of new observations and new constraints.

6.3. Results about Incremental Forgetting

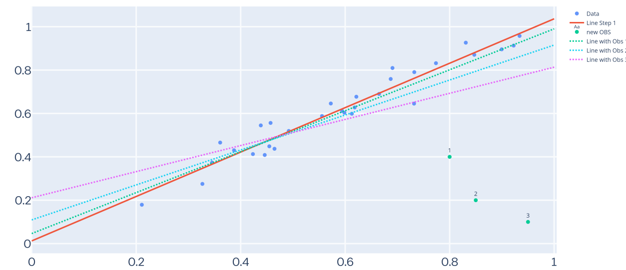
As regards Incremental Forgetting, in order not to burden the discussion, we did not insert any images, since in this part of the study we performed the reverse operations compared to those performed in incremental learning. In particular, we have made the following verification. Starting from the initial data, we plotted the regression line (in the linear case) or the parabola (in the quadratic case). Later, through the formula of incremental learning, we added an observation (in the first test) and a constraint (in the second test), thus finding two new equations. The correctness check of the incremental forgetting formula consisted in removing the observation (first test) and the constraint (second test) and checking that the resulting equations were consistent with the two initial equations, i.e. those without the addition of the observation nor of the bond. Having ascertained the equality of these equations, it was possible to deduce that both the incremental learning and the incremental forgetting formulas are correct.

6.4. Results of experiments in large size context

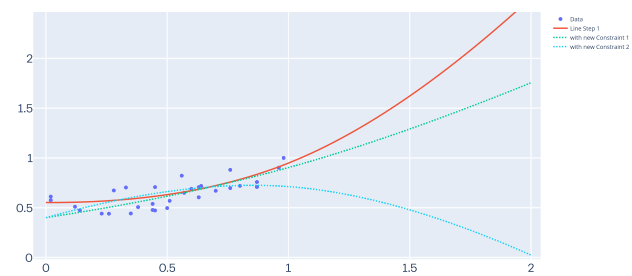
In this subsection we present the results we obtained by carrying out experiments in a high-dimensional context, that is in $\mathbb{R}^{10}$. We generated synthetic numbers, as mentioned



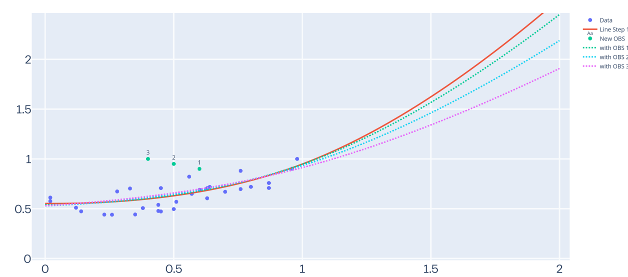
(a) Linear case: adding constraint $A_{\text{new}} = (1, 0)$, $b_{\text{new}} = (0.2)$



(b) Linear case: adding observations $x_1 = (0.8, 0.4)$, $x_2 = (0.85, 0.2)$, $x_3 = (0.95, 0.1)$



(c) Quadratic case: adding constraints $A_{\text{new1}} = (1, 0, 0)$, $b_{\text{new1}} = (0.4)$, $A_{\text{new2}} = (0, 0, 1)$, $b_{\text{new2}} = (-0.5)$



(d) Quadratic case: adding observations $x_1 = (0.6, 0.9)$, $x_2 = (0.5, 0.95)$, $x_3 = (0.4, 1)$

Figure 5: Incremental Learning

Step	x^9	x^8	x^7	x^6	x^5	x^4	x^3	x^2	x^1	x^0
1	1.00	1.99	3.03	3.98	5.02	5.91	7.08	8.01	8.97	10.01
2	-7554.32	35763.7	-71493.82	78536.06	-51619.85	20749.14	-4982.69	680.79	-35.13	11
3	-5149.71	24672.43	50029.05	55931.16	-37577.06	15539.59	-3872.32	558.89	-30	11
					...					
k	-25000	70000	-80000	60000	-35000	15000	-4000	580	-30	11

Table 1
Results about numerical experiments in high-dimensional context

before, in such a way that the final equation of the curve had the following structure:

$$w^k = x^9 + 2x^8 + 3x^7 + 4x^6 + 5x^5 + 6x^4 + 7x^3 + 8x^2 + 9x + 10$$

We present the results in numerical form. We applied the formula of the solution in closed form to these data and we found that indeed the solution found is what we expected, taking into account the negligible approximations. We can see the equation of this curve in the first row of Table 1. Then we added some observations and constraints, and the curve adjusts to those changes. Still in the Table 1, we can see how the line progressively evolves.

The added constraints are:

- $A_1 = (1, 0, 0, 0, 0, 0, 0, 0, 0, 0)$, $b_1 = 11$
- $A_2 = (0, 1, 0, 0, 0, 0, 0, 0, 0, 0)$, $b_2 = -30$
- $A_3 = (0, 0, 1, 0, 0, 0, 0, 0, 0, 0)$, $b_3 = 580$
- $A_4 = (0, 0, 0, 1, 0, 0, 0, 0, 0, 0)$, $b_4 = -4000$
- $A_5 = (0, 0, 0, 0, 1, 0, 0, 0, 0, 0)$, $b_5 = 15000$
- $A_6 = (0, 0, 0, 0, 0, 1, 0, 0, 0, 0)$, $b_6 = -35000$
- $A_7 = (0, 0, 0, 0, 0, 0, 1, 0, 0, 0)$, $b_7 = 60000$
- $A_8 = (0, 0, 0, 0, 0, 0, 0, 1, 0, 0)$, $b_8 = -80000$
- $A_9 = (0, 0, 0, 0, 0, 0, 0, 0, 1, 0)$, $b_9 = 70000$
- $A_{10} = (0, 0, 0, 0, 0, 0, 0, 0, 0, 1)$, $b_{10} = -25000$

We can see that the final equation satisfies all constraints. Even adding a further constraint, as long as we leave the problem feasible, and therefore making matrix A not full rank, the result remains consistent. For example, we have added the following constraint:

- $A_{11} = (7, -3, 1, 6, 9, -4, -1, -1, -5, -3)$, $b_{11} = -3253$

The results of Incremental Forgetting are not presented in order not to burden the discussion and since the equations of the curve, removing the constraints added in Incremental Learning, are the same.

7. Conclusions

In this letter we have presented a numerically computable closed formula for equally constrained quadratic programs, also deriving its incremental version for data stream processing, where both new data and new equality constraints can be added or removed over time.

In a future work, we will investigate Non-Archimedean iterative scheme for solving the batch problem in 4.6 using a pre-conditioned conjugate gradient approach. This would be a better solution when dealing with big data, especially if the involved matrices are sparse.

Acknowledgments

TBW TBW

References

[1] Benci, V., Cococcioni, M., Fiaschi, L., 2022. Non-standard analysis revisited: an easy axiomatic presentation oriented towards numerical applications. International Journal of Applied Mathematics and Computer Science 32.

[2] Bezanson, J., Edelman, A., Karpinski, S., Shah, V.B., 2017. Julia: A fresh approach to numerical computing. SIAM Review 59, 65–98. URL: https://doi.org/10.1137/141000671, doi:10.1137/141000671, arXiv:https://doi.org/10.1137/141000671.

[3] Cococcioni, M., Fiaschi, L., Lambertini, L., 2021. Non-archimedean zero-sum games. Journal of Computational and Applied Mathematics 393, 113483. URL: https://www.sciencedirect.com/science/article/pii/S0377042721001023, doi:https://doi.org/10.1016/j.cam.2021.113483.

[4] Escobar, L.A., Skarpness, B., 1984. A closed form solution for tre least squares regression problem with linear inequality constraints. Communications in Statistics - Theory and Methods 13, 1127–1134. URL: https://doi.org/10.1080/03610928408828746, doi:10.1080/03610928408828746, arXiv:https://doi.org/10.1080/03610928408828746.

[5] Gamage, S.S.U., Lasenby, J., 2002. New least squares solutions for estimating the average centre of rotation and the axis of rotation. Journal of Biomechanics 35, 87–93. URL: https://www.sciencedirect.com/science/article/pii/S0021929001001609, doi:https://doi.org/10.1016/S0021-9290(01)00160-9.

[6] Lawson, C.L., Hanson, R.J., 1995. Solving least squares problems. SIAM.

[7] Scott, J., Tuma, M., 2021. Solving large linear least squares problems with linear equality constraints. URL: https://arxiv.org/abs/2106.13142, doi:10.48550/ARXIV.2106.13142.