# Biased Regression Algorithms in the Quaternion Domain

Rosa María Fernández-Alcalá, José Domingo Jiménez-López, Jesús Navarro-Moreno, and Juan Carlos Ruiz-Molina

*Department of Statistics and Operations Research, University of Jaén, 23071 Jaén, Spain.*
*E-mails: {rmfernan,jdomingo,jnavarro,jcruiz}@ujaen.es. Tel:+34 953212729.*

**Abstract**

The ill-conditioned matrix problem in quaternion linear regression models is addressed in this paper and several dimension-reduction based regression methods for circumventing this problem are suggested. The algorithms are formulated in a general way and can be easily adapted to different scenarios: widely linear, semi-widely linear and strictly linear processing, in accordance with the properness properties presented by quaternion random vectors. A comparison with existing solutions is carried out by using both laboratory data and a color face database.

*Keywords:* Ill-conditioned Matrices, Partial Least Squares, Principal Covariates Regression, Properness Properties, Quaternion Regression Models.

## 1. Introduction

Quaternions have received increasingly widespread attention because of their successful applications in signal and image processing, 3D rotation representation, robotics, and computer graphics [1]. The advantages of quaternion representation over other domains lie in the geometric insights offered by its algebra and the fact that it can be used to formulate compact and explicit models for the data at hand.

One of the most outstanding problems in signal and image processing is that of estimation. Linear regression is one of the most widely used quaternion

estimation methods for the prediction of one dataset from another. Quaternion linear regression (QLR) has been applied in color face recognition [2], [3], [4], [5], [6], [7], [8] and in attitude estimation [9].

The general approach used for estimating QLR models is ordinary least squares (OLS) regression. However, OLS regression can suffer from the negative effects of the ill-conditioned matrix problem. Ill-conditioned matrices lead to numerical instability when performing calculations, such as matrix inversion, because small changes in the input data or rounding errors can result in disproportionately large changes in the results. Several factors can contribute to a matrix being ill-conditioned: scale disparity, nearly dependent columns (or rows), a sample size smaller than the number of variables, etc.

Biased regression methods (BRMs) are used as an alternative approach to OLS regression in ill-conditioned linear models. These methods aim to stabilize the parameter estimates and provide more reliable predictions. There are several types of BRMs in the literature: quaternion ridge-regression [2], [3], [4], quaternion principal component regression (QPCR) [3], [5], [7], quaternion singular value decomposition regression (QSVDR) [10], and quaternion partial least squares regression (QPLSR) [8].

QSVDR, QPCR, and QPLSR are all dimension-reduction based regression methods that involve two modeling aspects: reducing the predictors and predicting the response variables. The reasoning behind these techniques is that the data observed are generated by a system which is driven by a small number of components or latent variables (which have not been directly observed or measured). Moreover, all these methods operate in the same way: they aim to reduce the independent variables to just a few latent variables, and then regress the response variables on the components rather than the predictors. This simplifies matters because far fewer regression coefficients have to be estimated, and the components obtained are also uncorrelated (except in the QSVD regression), which deals with the issues surrounding ill-conditioning effects.

It is well-known that the structure of the data is an important aspect to be considered when selecting the most appropriate biased regression algorithm

in practice [8]. Since there is no solution that is superior to all the others in every experimental situation, the above decision tends to be made in an *ad hoc* manner. Therefore, this paper proposes some alternative BRMs in the quaternion domain to widen the variety of options available to users. Specifically, four new quaternion methods are given: QPLS-W2A, QPLS2, QSIMPLS, and QPCovR. Firstly, their theoretical properties are studied and then, their algorithmic versions are provided. They are proposed in a general form, which is easily adaptable to different scenarios: WL, semi-widely (SWL), or strictly linear (SL) regressions, with the latter two methods being used when properness conditions hold [10]. SWL and SL regressions are interesting because they reduce the computational burden involved [11]. Additionally, the algorithms proposed are compared on synthetic datasets using the solutions that already exist in the literature, and some practical recommendations are given. Finally, we illustrate the practical application of the BRMs in quaternion linear regression classification (QLRC).

The paper is organized as follows. Section 2 gives a brief review of quaternions and some of their properties. Section 3 describes the main dimension-reduction based regression methods existing in the literature, and introduces the novel quaternion regression algorithms. In Section 4, several numerical simulations compare the performance of the algorithms suggested with the existing solutions and recommendations for application are given. The utility of the provided results is demonstrated with an application for color image classification in Section 5. Conclusions are given in Section 6. To ensure continuity, all proofs are included in the Appendix.

## 2. Preliminaries

The aim of this section is to introduce the notation and basic concepts used in this paper.

### 2.1. Notation

Throughout this paper, all the random variables are assumed to have zero-mean. We use boldfaced uppercase letters to denote matrices, boldfaced lowercase letters for column vectors, and lightfaced letters for scalar quantities. For some special matrices we will also use boldfaced uppercase italicized letters (e.g., $\boldsymbol{\mathcal{T}}$). The superscripts "$*$", "$\mathtt{T}$" and "$\mathtt{H}$" represent the quaternion conjugate, transpose, and Hermitian (conjugate transpose), respectively. $\mathbf{I}_m$ denotes the identity matrix of dimension $m$ and $\mathbf{0}$ is a matrix zero with suitable dimensions. The notation $\mathbf{A} \in \mathbb{R}^{m\times n}$ (respectively $\mathbf{A} \in \mathbb{Q}^{m\times n}$) means that $\mathbf{A}$ is a real (respectively quaternion) $m \times n$ matrix. We will remove one dimension in the case of vectors (e.g., $\mathbf{r} \in \mathbb{Q}^m$ means that $\mathbf{r}$ is a $m$-dimensional quaternion vector).

$E[\cdot]$ is the expectation operator, $\mathrm{Tr}(\cdot)$ is the trace of a matrix, $\mathrm{diag}(\cdot)$ is a diagonal matrix with the elements specified on the main diagonal, and the symbol $\perp$ denotes orthogonality. Finally, "$\otimes$" denotes the Kronecker product.

### 2.2. Quaternion algebra

The set of real quaternions is defined by

$$\mathbb{Q} := \{x = x_0 + x_1\mathtt{i} + x_2\mathtt{j} + x_3\mathtt{k} : x_0, x_1, x_2, x_3 \in \mathbb{R}\} \tag{1}$$

where $\mathtt{i}$, $\mathtt{j}$ and $\mathtt{k}$ are three imaginary units. Quaternions form a division algebra when equipped with the componentwise addition, the componentwise scalar multiplication over $\mathbb{R}$, and the Hamilton product given by $\mathtt{i}^2 = \mathtt{j}^2 = \mathtt{k}^2 = \mathtt{ijk} = -1$, $\mathtt{ij} = -\mathtt{ji} = \mathtt{k}$, $\mathtt{jk} = -\mathtt{kj} = \mathtt{i}$, and $\mathtt{ki} = -\mathtt{ik} = \mathtt{j}$. Obviously, quaternion multiplication cannot satisfy the commutative law. The conjugate of $x$ is $x^* = x_0 - x_1\mathtt{i} - x_2\mathtt{j} - x_3\mathtt{k}$ and its modulus is $|x| = \sqrt{x_0^2 + x_1^2 + x_2^2 + x_3^2}$.

Three special cases of involutions can be defined as

$$\begin{aligned} x^{\mathtt{i}} &:= -\mathtt{i}x\mathtt{i} = x_0 + x_1\mathtt{i} - x_2\mathtt{j} - x_3\mathtt{k} \\ x^{\mathtt{j}} &:= -\mathtt{j}x\mathtt{j} = x_0 - x_1\mathtt{i} + x_2\mathtt{j} - x_3\mathtt{k} \\ x^{\mathtt{k}} &:= -\mathtt{k}x\mathtt{k} = x_0 - x_1\mathtt{i} - x_2\mathtt{j} + x_3\mathtt{k} \end{aligned} \tag{2}$$

Let $\mathbb{Q}^p$ be the quaternion vector space of dimension $p$ over the field of quaternions $\mathbb{Q}$. Over this vector space, a scalar product can be defined as

$$\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^{\mathtt{H}} \mathbf{y} = \sum_{i=1}^{p} x_i^* y_i \tag{3}$$

with $\mathbf{x} = [x_1, \ldots, x_p]^{\mathtt{T}}$ and $\mathbf{y} = [y_1, \ldots, y_p]^{\mathtt{T}}$. The norm of $\mathbf{x} \in \mathbb{Q}^p$ is defined as $\|\mathbf{x}\| = \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle}$.

Denote a quaternion matrix $\mathbf{A} \in \mathbb{Q}^{m \times p}$ as $\mathbf{A} = \mathbf{A}_0 + \mathbf{A}_1\mathtt{i} + \mathbf{A}_2\mathtt{j} + \mathbf{A}_3\mathtt{k}$, where $\mathbf{A}_0, \mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3 \in \mathbb{R}^{m \times p}$. A matrix $\mathbf{A} \in \mathbb{Q}^{p \times p}$ is said to be Hermitian if $\mathbf{A} = \mathbf{A}^{\mathtt{H}}$ and unitary if $\mathbf{A}^{\mathtt{H}}\mathbf{A} = \mathbf{I}_p$. An inner product can be defined as $\langle \mathbf{A}, \mathbf{B} \rangle = \mathrm{Tr}(\mathbf{B}^{\mathtt{H}}\mathbf{A})$, with $\mathbf{A}, \mathbf{B} \in \mathbb{Q}^{m \times p}$. Moreover, the norm $\|\mathbf{A}\| = \sqrt{\langle \mathbf{A}, \mathbf{A} \rangle}$ is the Frobenius norm. General properties of matrices over $\mathbb{Q}$ can be found in [12]. The Moore-Penrose inverse $\mathbf{A}^+ \in \mathbb{Q}^{p \times m}$ of a matrix $\mathbf{A} \in \mathbb{Q}^{m \times p}$ always exists, it is unique and if $\mathbf{A} \in \mathbb{Q}^{p \times p}$ is invertible, then $\mathbf{A}^+ = \mathbf{A}^{-1}$ [13]. Moreover, matrix $\mathbf{\Pi_A} := \mathbf{A}\mathbf{A}^+$ is the orthogonal projector onto the column space of $\mathbf{A}$ [14], denoted by $\mathcal{C}(\mathbf{A})$.

Assume $\mathbf{A} \in \mathbb{Q}^{p \times p}$ is Hermitian, then $\mathbf{A}$ has exactly $p$ real right eigenvalues, and there is a unitary matrix $\mathbf{U} \in \mathbb{Q}^{p \times p}$ such that

$$\mathbf{A} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{\mathtt{H}} \tag{4}$$

The QSVD of a matrix $\mathbf{A} \in \mathbb{Q}^{m \times p}$ with rank $r$ is

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathtt{H}} = \sum_{i=1}^{r} \mathbf{u}_i \mathbf{v}_i^{\mathtt{H}} \sigma_i \tag{5}$$

where the singular values are $\sigma_i > 0$ [12]. Moreover, $\mathbf{A}_s = \sum_{i=1}^{s} \mathbf{u}_i \mathbf{v}_i^{\mathtt{H}} \sigma_i$ is the best rank-$s$ ($s < r$) approximation of $\mathbf{A}$ [15].

Define the real adjoint matrix of $\mathbf{A} = \mathbf{A}_0 + \mathbf{A}_1\mathtt{i} + \mathbf{A}_2\mathtt{j} + \mathbf{A}_3\mathtt{k} \in \mathbb{Q}^{m \times p}$ as

$$\mathbf{\Phi}(\mathbf{A}) = \begin{pmatrix} \mathbf{A}_0 & -\mathbf{A}_1 & -\mathbf{A}_2 & -\mathbf{A}_3 \\ \mathbf{A}_1 & \mathbf{A}_0 & -\mathbf{A}_3 & \mathbf{A}_2 \\ \mathbf{A}_2 & \mathbf{A}_3 & \mathbf{A}_0 & -\mathbf{A}_1 \\ \mathbf{A}_3 & -\mathbf{A}_2 & \mathbf{A}_1 & \mathbf{A}_0 \end{pmatrix} \in \mathbb{R}^{4m \times 4p} \tag{6}$$

The next properties of real adjoint matrices have been established [4], [16].

**Property 1.** *Let* $\mathbf{A}, \mathbf{B} \in \mathbb{Q}^{m\times p}$, $\mathbf{C} \in \mathbb{Q}^{p\times m}$, *and* $\mathbf{D} \in \mathbb{Q}^{p\times p}$. *Then,*

1. $\mathbf{A} = \mathbf{B}$ *if and only if* $\mathbf{\Phi}(\mathbf{A}) = \mathbf{\Phi}(\mathbf{B})$.
2. $\mathbf{\Phi}(\mathbf{A}+\mathbf{B}) = \mathbf{\Phi}(\mathbf{A}) + \mathbf{\Phi}(\mathbf{B})$ *and* $\mathbf{\Phi}(\mathbf{AC}) = \mathbf{\Phi}(\mathbf{A})\mathbf{\Phi}(\mathbf{C})$.
3. $\mathbf{\Phi}(\lambda\mathbf{A}) = \mathbf{\Phi}(\mathbf{A}\lambda) = \lambda\mathbf{\Phi}(\mathbf{A})$, *with* $\lambda \in \mathbb{R}$.
4. $\mathbf{\Phi}(\mathbf{A}^{\mathtt{H}}) = \mathbf{\Phi}^{\mathtt{T}}(\mathbf{A})$. *Thus,* $\mathbf{D}$ *is Hermitian if and only if* $\mathbf{\Phi}(\mathbf{D})$ *is symmetric.*
5. $\mathrm{rank}(\mathbf{A}) = \frac{1}{4}\,\mathrm{rank}\,(\mathbf{\Phi}(\mathbf{A}))$.
6. $\mathbf{\Phi}(\mathbf{D}^{-1}) = \big(\mathbf{\Phi}(\mathbf{D})\big)^{-1}$ *and* $\mathbf{\Phi}(\mathbf{A}^{+}) = \mathbf{\Phi}^{+}(\mathbf{A})$.
7. $\|\mathbf{\Phi}(\mathbf{A})\|^2 = 4\|\mathbf{A}\|^2$, *where the first norm is the Frobenius norm in* $\mathbb{R}^{4m\times 4p}$.
8. $\mathbf{D}$ *has decomposition* (4) *if and only if* $\mathbf{\Phi}(\mathbf{D}) = \mathbf{\Phi}(\mathbf{U})\mathbf{\Phi}(\mathbf{\Lambda})\mathbf{\Phi}^{\mathtt{T}}(\mathbf{U})$ *is an eigendecomposition of* $\mathbf{\Phi}(\mathbf{D})$. *As a consequence, if* $\mathbf{D}$ *is also positive semidefinite then,* $\mathbf{\Phi}(\mathbf{D}^{1/2}) = (\mathbf{\Phi}(\mathbf{D}))^{1/2}$.

In operations using the real adjoint matrix, employing structure-preserving arithmetic rules often saves time when compared to general operations [17], [18].

### *2.3. Quaternion widely linear processing*

Consider a random vector $\mathbf{x} = \mathbf{x}_0+\mathbf{x}_1\mathtt{i}+\mathbf{x}_2\mathtt{j}+\mathbf{x}_3\mathtt{k} \in \mathbb{Q}^p$, with $\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3 \in \mathbb{R}^p$. We define the augmented vector of $\mathbf{x}$ as $\bar{\mathbf{x}} = [\mathbf{x}^{\mathtt{T}}, \mathbf{x}^{\mathtt{iT}}, \mathbf{x}^{\mathtt{jT}}, \mathbf{x}^{\mathtt{kT}}]^{\mathtt{T}}$. The following relationship between the augmented vector of $\mathbf{x}$ and the real vector $\mathbf{x}_{re} = [\mathbf{x}_0^{\mathtt{T}}, \mathbf{x}_1^{\mathtt{T}}, \mathbf{x}_2^{\mathtt{T}}, \mathbf{x}_3^{\mathtt{T}}]^{\mathtt{T}}$ is satisfied: $\bar{\mathbf{x}} = \boldsymbol{\mathcal{T}}_p\mathbf{x}_{re}$, where

$$\boldsymbol{\mathcal{T}}_p = \begin{bmatrix} 1 & \mathtt{i} & \mathtt{j} & \mathtt{k} \\ 1 & \mathtt{i} & -\mathtt{j} & -\mathtt{k} \\ 1 & -\mathtt{i} & \mathtt{j} & -\mathtt{k} \\ 1 & -\mathtt{i} & -\mathtt{j} & \mathtt{k} \end{bmatrix} \otimes \mathbf{I}_p \tag{7}$$

with $\boldsymbol{\mathcal{T}}_p^{\mathtt{H}}\boldsymbol{\mathcal{T}}_p = 4\mathbf{I}_{4p}$. The covariance matrix of $\bar{\mathbf{x}}$, called augmented covariance matrix, is given by

$$\mathrm{E}[\bar{\mathbf{x}}\bar{\mathbf{x}}^{\mathtt{H}}] = \begin{pmatrix} \mathrm{E}[\mathbf{x}\mathbf{x}^{\mathtt{H}}] & \mathrm{E}[\mathbf{x}\mathbf{x}^{\mathtt{iH}}] & \mathrm{E}[\mathbf{x}\mathbf{x}^{\mathtt{jH}}] & \mathrm{E}[\mathbf{x}\mathbf{x}^{\mathtt{kH}}] \\ (\mathrm{E}[\mathbf{x}\mathbf{x}^{\mathtt{iH}}])^{\mathtt{i}} & (\mathrm{E}[\mathbf{x}\mathbf{x}^{\mathtt{H}}])^{\mathtt{i}} & (\mathrm{E}[\mathbf{x}\mathbf{x}^{\mathtt{kH}}])^{\mathtt{i}} & (\mathrm{E}[\mathbf{x}\mathbf{x}^{\mathtt{jH}}])^{\mathtt{i}} \\ (\mathrm{E}[\mathbf{x}\mathbf{x}^{\mathtt{jH}}])^{\mathtt{j}} & (\mathrm{E}[\mathbf{x}\mathbf{x}^{\mathtt{kH}}])^{\mathtt{j}} & (\mathrm{E}[\mathbf{x}\mathbf{x}^{\mathtt{H}}])^{\mathtt{j}} & (\mathrm{E}[\mathbf{x}\mathbf{x}^{\mathtt{iH}}])^{\mathtt{j}} \\ (\mathrm{E}[\mathbf{x}\mathbf{x}^{\mathtt{kH}}])^{\mathtt{k}} & (\mathrm{E}[\mathbf{x}\mathbf{x}^{\mathtt{jH}}])^{\mathtt{k}} & (\mathrm{E}[\mathbf{x}\mathbf{x}^{\mathtt{iH}}])^{\mathtt{k}} & (\mathrm{E}[\mathbf{x}\mathbf{x}^{\mathtt{H}}])^{\mathtt{k}} \end{pmatrix} \tag{8}$$

In linear mean-squared error (MSE) estimation, to explain completely the second-order information of $\mathbf{x}$, it is necessary to consider the augmented statistics, $\bar{\mathbf{x}}$ and $\mathrm{E}[\bar{\mathbf{x}}\bar{\mathbf{x}}^{\mathtt{H}}]$, giving rise to the so-called WL processing [10, 19]. Specifically, we have to operate on the four involutions simultaneously and define the linear transformations

$$\mathbf{y}^{\mathtt{T}} = \bar{\mathbf{x}}^{\mathtt{T}}\mathbf{B} = \mathbf{x}^{\mathtt{T}}\mathbf{B}_1 + \mathbf{x}^{\mathtt{i}\mathtt{T}}\mathbf{B}_2 + \mathbf{x}^{\mathtt{j}\mathtt{T}}\mathbf{B}_3 + \mathbf{x}^{\mathtt{k}\mathtt{T}}\mathbf{B}_4 \tag{9}$$

with $\mathbf{B} = [\mathbf{B}_1^{\mathtt{T}}, \mathbf{B}_2^{\mathtt{T}}, \mathbf{B}_3^{\mathtt{T}}, \mathbf{B}_4^{\mathtt{T}}]^{\mathtt{T}} \in \mathbb{Q}^{4p\times q}$. The above equation can be written as

$$\bar{\mathbf{y}}^{\mathtt{T}} = \bar{\mathbf{x}}^{\mathtt{T}}\bar{\mathbf{B}} \tag{10}$$

where

$$\bar{\mathbf{B}} = \begin{pmatrix} \mathbf{B}_1 & \mathbf{B}_2^{\mathtt{i}} & \mathbf{B}_3^{\mathtt{j}} & \mathbf{B}_4^{\mathtt{k}} \\ \mathbf{B}_2 & \mathbf{B}_1^{\mathtt{i}} & \mathbf{B}_4^{\mathtt{j}} & \mathbf{B}_3^{\mathtt{k}} \\ \mathbf{B}_3 & \mathbf{B}_4^{\mathtt{i}} & \mathbf{B}_1^{\mathtt{j}} & \mathbf{B}_2^{\mathtt{k}} \\ \mathbf{B}_4 & \mathbf{B}_3^{\mathtt{i}} & \mathbf{B}_2^{\mathtt{j}} & \mathbf{B}_1^{\mathtt{k}} \end{pmatrix} \in \mathbb{Q}^{4p\times 4q} \tag{11}$$

Several concepts of quaternion properness can be defined by means of the cancelation of the complementary covariance matrices: $\mathrm{E}[\mathbf{x}\mathbf{x}^{\mathtt{i}\mathtt{H}}]$, $\mathrm{E}[\mathbf{x}\mathbf{x}^{\mathtt{j}\mathtt{H}}]$, and $\mathrm{E}[\mathbf{x}\mathbf{x}^{\mathtt{k}\mathtt{H}}]$ [10]. In particular, a random vector $\mathbf{x} \in \mathbb{Q}^p$ is $\mathbb{C}^{\mathtt{i}}$-proper if and only if $\mathrm{E}[\mathbf{x}\mathbf{x}^{\mathtt{j}\mathtt{H}}] = \mathrm{E}[\mathbf{x}\mathbf{x}^{\mathtt{k}\mathtt{H}}] = \mathbf{0}$. In such a case, the augmented covariance matrix is given by

$$\mathrm{E}[\bar{\mathbf{x}}\bar{\mathbf{x}}^{\mathtt{H}}] = \begin{pmatrix} \mathrm{E}[\tilde{\mathbf{x}}\tilde{\mathbf{x}}^{\mathtt{H}}] & \mathbf{0} \\ \mathbf{0} & (\mathrm{E}[\tilde{\mathbf{x}}\tilde{\mathbf{x}}^{\mathtt{H}}])^{\mathtt{j}} \end{pmatrix} \tag{12}$$

where $\tilde{\mathbf{x}} = [\mathbf{x}^{\mathtt{T}}, \mathbf{x}^{\mathtt{i}\mathtt{T}}]^{\mathtt{T}}$ and the adequate processing is called SWL, which is characterized by the linear transformation

$$\mathbf{y}^{\mathtt{T}} = \tilde{\mathbf{x}}^{\mathtt{T}}\tilde{\mathbf{B}} = \mathbf{x}^{\mathtt{T}}\mathbf{B}_1 + \mathbf{x}^{\mathtt{i}\mathtt{T}}\mathbf{B}_2 \tag{13}$$

with $\tilde{\mathbf{B}} = [\mathbf{B}_1^{\mathtt{T}}, \mathbf{B}_2^{\mathtt{T}}]^{\mathtt{T}}$. Finally, $\mathbf{x}$ is $\mathbb{Q}$-proper if and only if $\mathrm{E}[\mathbf{x}\mathbf{x}^{\mathtt{i}\mathtt{H}}] = \mathrm{E}[\mathbf{x}\mathbf{x}^{\mathtt{j}\mathtt{H}}] = \mathrm{E}[\mathbf{x}\mathbf{x}^{\mathtt{k}\mathtt{H}}] = \mathbf{0}$. $\mathbb{Q}$-properness is characterized by a SL processing, i.e., the suitable linear transformations are now

$$\mathbf{y}^{\mathtt{T}} = \mathbf{x}^{\mathtt{T}}\mathbf{B}_1 \tag{14}$$

Properness properties provide a significant saving in the computational burden of the processing associated (SWL o SL), when compared to the WL processing [11], since the dimensions of the vectors and matrices involved is reduced by half ($\mathbb{C}^{\mathtt{i}}$-proper case) or a quarter ($\mathbb{Q}$-proper case).

## 3. Biased regression: dimension-reduction based methods

Consider the general setting of a quaternion multivariate linear regression (QMLR) to model the relation between two data sets. Denote $\mathbf{X} \in \mathbb{Q}^{n\times p}$ and $\mathbf{Y} \in \mathbb{Q}^{n\times q}$ the matrices containing the random samples of size $n$ from two random vectors $\mathbf{x} \in \mathbb{Q}^p$ and $\mathbf{y} \in \mathbb{Q}^q$, respectively. The problem of QMLR is formulated as follows. Given the linear model

$$\mathbf{Y} = \mathbf{X}\mathbf{B} + \mathbf{\Omega} \tag{15}$$

where $\mathbf{B} \in \mathbb{Q}^{p\times q}$ is the regression coefficients matrix and $\mathbf{\Omega}$ is a noise matrix, the aim is to solve the following linear least squares estimation problem:

$$\min_{\mathbf{B}} \|\mathbf{Y} - \mathbf{X}\mathbf{B}\| \tag{16}$$

The solution to this problem is well-known and is given by the OLS estimator: $\hat{\mathbf{B}}_{\mathrm{OLS}} = \mathbf{X}^{+}\mathbf{Y} = (\mathbf{X}^{\mathtt{H}}\mathbf{X})^{-1}\mathbf{X}^{\mathtt{H}}\mathbf{Y}$, when $\mathbf{X}^{\mathtt{H}}\mathbf{X}$ is nonsingular. Then, we can project $\mathbf{Y}$ onto the range of $\mathbf{X}$ to obtain the prediction of $\mathbf{Y}$ as $\hat{\mathbf{Y}} = \mathbf{\Pi}_{\mathbf{X}}\mathbf{Y} = \mathbf{X}\hat{\mathbf{B}}_{\mathrm{OLS}}$. However, when $\mathbf{X}^{\mathtt{H}}\mathbf{X}$ is nearly singular, it is possible to find rather large weights in $\hat{\mathbf{B}}_{\mathrm{OLS}}$ and thus, unreliable values in $\hat{\mathbf{Y}}$. This problem is the so-called bouncing beta problem[1]. Actually, the accurate computation of $\hat{\mathbf{B}}_{\mathrm{OLS}}$ can be a challenge for ill-conditioned matrices $\mathbf{X}$.

Several approaches have been developed to cope with the ill-conditioned matrix problem in QMLR. A popular method used to avoid the bouncing beta problem is QPCR [3], [5], [7]. In QPCR, the predictor variables are summarized

[1] The bouncing beta problem also means that removing or adding a single observation can already alter the regression weights drastically.

by means of a limited number of components, which are then used as the independent variables of a regression model for predicting $\mathbf{Y}$. Specifically, in the first stage, a PCA is carried out on $\mathbf{X}$, which boils down to minimizing the loss function $\|\mathbf{X}-\mathbf{T}_r\mathbf{P}_r^{\mathtt{H}}\|$ over arbitrary $\mathbf{T}_r \in \mathbb{Q}^{n\times r}$ and $\mathbf{P}_r \in \mathbb{Q}^{p\times r}$ matrices, where the number of components $r$ (i.e., the number of columns in $\mathbf{T}_r = [\mathbf{t}_1, \ldots, \mathbf{t}_r]$) is usually much smaller than the rank of data matrix $\mathbf{X}$. It is well known that the solution for this is obtained from the QSVD of $\mathbf{X}$ given by (5), by taking $\mathbf{T}_r$ proportional to the first $r$ left singular vectors of $\mathbf{X}$. The columns of $\mathbf{T}_r$ now contain $r$ linear combinations of the columns of $\mathbf{X}$, $\mathbf{T}_r = \mathbf{X}\mathbf{W}_r$ with $\mathbf{W}_r \in \mathbb{Q}^{p\times r}$, and the matrix of loadings is obtained as $\mathbf{P}_r^{\mathtt{H}} = \mathbf{T}_r^{+}\mathbf{X}$. Next, the summarized scores in $\mathbf{T}_r$ are used for the prediction of $\mathbf{Y}$-scores. That is, in the next step in QPCR, $\mathbf{Y}$-scores are regressed on $\mathbf{T}$-scores by minimizing $\|\mathbf{Y} - \mathbf{T}_r\mathbf{Q}_r^{\mathtt{H}}\|$, where the solution is given by $\mathbf{Q}_r^{\mathtt{H}} = \mathbf{T}_r^{+}\mathbf{Y}$. Finally, the predictions of $\mathbf{Y}$ are given by $\hat{\mathbf{Y}} = \mathbf{\Pi}_{\mathbf{T}_r}\mathbf{Y} = \mathbf{X}\hat{\mathbf{B}}_{\mathrm{PCR}}$, with $\hat{\mathbf{B}}_{\mathrm{PCR}} = \mathbf{W}_r\mathbf{T}_r^{+}\mathbf{Y}$.

QPCR works well because the orthogonality of the $\mathbf{t}_i$ components eliminates the ill-conditioned matrix problem. However, the problem of choosing an optimum subset of predictors remains. One possible strategy is to keep only a few of the first components. But these components were originally chosen to explain $\mathbf{X}$ rather than $\mathbf{Y}$, and so, nothing guarantees that the principal components will be relevant for the prediction of $\mathbf{Y}$.

Alternatively, QPLS methods can be applied. QPLS is a dimension reduction technique similar to QPCR in that the matrix of regressors is replaced by a reduced set of linear combinations, $\mathbf{T}_r$. There is, however, one essential difference: in QPCR the linear combinations are derived without reference to the response variables $\mathbf{Y}$, but in QPLS the responses observed play an essential role.

In its more general form, QPLS regression creates two sets of components, one for the explanatory variables, $\mathbf{T}_r = \mathbf{X}\mathbf{W}_r \in \mathbb{Q}^{n\times r}$, and another for the response variables, $\mathbf{S}_r = \mathbf{Y}\mathbf{C}_r \in \mathbb{Q}^{n\times r}$. To this end, QPLS operates under the assumption of a basic latent decomposition of the response and predictor

matrices

$$\begin{aligned} \mathbf{Y} &= \mathbf{S}_r \mathbf{Q}_r^{\mathtt{H}} + \mathbf{F}^{(r)} \\ \mathbf{X} &= \mathbf{T}_r \mathbf{P}_r^{\mathtt{H}} + \mathbf{E}^{(r)} \end{aligned} \tag{17}$$

where $\mathbf{Q}_r \in \mathbb{Q}^{q\times r}$ and $\mathbf{P}_r \in \mathbb{Q}^{p\times r}$ represent matrices of loadings, and $\mathbf{F}^{(r)} \in \mathbb{Q}^{n\times q}$ and $\mathbf{E}^{(r)} \in \mathbb{Q}^{n\times p}$ are the matrices of residuals. This general approach models the relations between the two blocks of data in a symmetrical way. To specify the latent component matrices $\mathbf{S}_r$ and $\mathbf{T}_r$ when using QPLS, the columns of $\mathbf{C}_r = [\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_r]$ and the columns of $\mathbf{W}_r = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_r]$ must be found from successive iterations. The choice of the appropriate number $r$ of components is an essential task.

In this section, different QPLS methods to address ill-conditioned matrix problems in quaternion regression are analyzed. Firstly, the QPLS-SB algorithm proposed in [10] is described. After that, four new algorithms, named QPLS-W2A, QPLS2, QSIMPLS, and QPCovR, are designed in a general setting for use in the most general case of improper random vectors, as well as in cases where the signal is $\mathbb{C}^{\mathtt{i}}$-proper or $\mathbb{Q}$-proper. Moreover, for the purposes of comparison, the WL-QPLS devised in [8] is reviewed.

### *3.1. QPLS-SB algorithm*

The first form of QPLS that we consider is based on the QSVD of $\mathbf{X}^{\mathtt{H}}\mathbf{Y}$ and was studied in [10]. Specifically, if $\mathbf{X}^{\mathtt{H}}\mathbf{Y} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathtt{H}}$, then we select $\mathbf{T}_r = \mathbf{X}\mathbf{U}_r$ and $\mathbf{S}_r = \mathbf{Y}\mathbf{V}_r$. This version satisfies that $\mathbf{T}_r^{\mathtt{H}}\mathbf{S}_r$ is a diagonal matrix, but $\langle \mathbf{t}_i, \mathbf{t}_j \rangle \neq 0$ and $\langle \mathbf{s}_i, \mathbf{s}_j \rangle \neq 0$, $i \neq j$, with $\mathbf{t}_i$ and $\mathbf{s}_i$ the columns of $\mathbf{T}_r$ and $\mathbf{S}_r$, respectively. In some cases, this non-orthogonality of the latent vectors does not eliminate the ill-conditioned matrix problem, and it also does not provide the most parsimonious regresssion model [8]. The prediction of the $\mathbf{Y}$-scores is obtained as: $\hat{\mathbf{Y}} = \mathbf{\Pi}_{\mathbf{T}_r}\mathbf{Y} = \mathbf{X}\hat{\mathbf{B}}_{\mathrm{SB}}$, with $\hat{\mathbf{B}}_{\mathrm{SB}} = \mathbf{U}_r\mathbf{T}_r^{+}\mathbf{Y}$.

### *3.2. QPLS-W2A algorithm*

The non-orthogonality of the latent vectors in QPLS-SB can be circumvented by applying the NIPALS algorithm [20]. The NIPALS algorithm has previously

been used in the quaternion domain in [8] and we will analyze this QPLS solution below. At this point, we use NIPALS under the assumption that decomposition (17) holds to devise a new technique called QPLS-W2A. The following result shows their main characteristics.

**Theorem 1.** *Let* $\mathbf{X}^{(1)} = \mathbf{X}$ *and* $\mathbf{Y}^{(1)} = \mathbf{Y}$*, and define*

$$\mathbf{X}^{(j+1)} = \mathbf{X}^{(j)} - \mathbf{t}_j \mathbf{p}_j^{\mathtt{H}} \qquad \mathbf{Y}^{(j+1)} = \mathbf{Y}^{(j)} - \mathbf{s}_j \mathbf{q}_j^{\mathtt{H}}, \qquad j = 1, \ldots, r \tag{18}$$

*where* $\mathbf{t}_j = \mathbf{X}^{(j)}\mathbf{u}_j$*,* $\mathbf{s}_j = \mathbf{Y}^{(j)}\mathbf{v}_j$*,* $(\mathbf{u}_j, \mathbf{v}_j)$ *is the first pair of singular vectors of* $\mathbf{X}^{(j)\mathtt{H}}\mathbf{Y}^{(j)}$*,* $\mathbf{p}_j^{\mathtt{H}} = \mathbf{t}_j^{+}\mathbf{X}^{(j)}$ *and* $\mathbf{q}_j^{\mathtt{H}} = \mathbf{s}_j^{+}\mathbf{Y}^{(j)}$*. Then,*

1. $\mathbf{u}_i \perp \mathbf{u}_j$ *and* $\mathbf{v}_i \perp \mathbf{v}_j$*,* $i \neq j$*.*
2. $\mathbf{t}_i \perp \mathbf{t}_j$ *and* $\mathbf{s}_i \perp \mathbf{s}_j$*,* $i \neq j$*.*
3. $\mathbf{T}_j = \mathbf{X}\mathbf{R}_j$*,* $j = 1, \ldots, r$*, with* $\mathbf{R}_j = \mathbf{U}_j(\mathbf{P}_j^{\mathtt{H}}\mathbf{U}_j)^{-1}$*,* $\mathbf{R}_j = [\mathbf{r}_1, \ldots, \mathbf{r}_j]$*,* $\mathbf{U}_j = [\mathbf{u}_1, \ldots, \mathbf{u}_j]$ *and* $\mathbf{P}_j = [\mathbf{p}_1, \ldots, \mathbf{p}_j]$*. A similar expression can be derived for* $\mathbf{S}_j$*.*
4. $\mathbf{X}^{(j+1)} = \mathbf{\Pi}_{\mathbf{T}_j^{\perp}}\mathbf{X}$ *and* $\mathbf{Y}^{(j+1)} = \mathbf{\Pi}_{\mathbf{S}_j^{\perp}}\mathbf{Y}$*.*
5. $\hat{\mathbf{Y}} = \mathbf{\Pi}_{\mathbf{T}_r}\mathbf{Y} = \mathbf{X}\hat{\mathbf{B}}_{\mathrm{W2A}}$ *with* $\hat{\mathbf{B}}_{\mathrm{W2A}} = \mathbf{R}_r\mathbf{T}_r^{+}\mathbf{Y} = \mathbf{U}_r(\mathbf{X}\mathbf{U}_r)^{+}\mathbf{Y}$*.*

Proof. See Appendix A.1.

The following observations are in order:

1. Notice that $(\mathbf{u}_j, \mathbf{v}_j)$ are the first singular vectors of different matrices ($\mathbf{X}^{(j)\mathtt{H}}\mathbf{Y}^{(j)}$ changes with $j$) and thus, they provide the best rank-one approximation of $\mathbf{X}^{(j)\mathtt{H}}\mathbf{Y}^{(j)}$ at each iteration of the procedure.
2. Theorem 1-4 shows how the NIPALS algorithm operates: by projecting the columns of $\mathbf{X}$ onto the space orthogonal to the $\mathbf{t}_j$ already extracted, ensuring the orthogonality of the latent components.
3. From Theorem 1-3, it follows that $\mathbf{t}_j \in \mathcal{C}(\mathbf{X})$. Thus, $\{\mathbf{t}_1, \ldots, \mathbf{t}_r\}$ is an orthogonal basis of a subspace of dimension $r$ of $\mathcal{C}(\mathbf{X})$.
4. The following recursion also holds: $\mathbf{r}_j = \mathbf{N}^{(j)}\mathbf{u}_j$, with $\mathbf{N}^{(1)} = \mathbf{I}_p$ and $\mathbf{N}^{(j)} = \mathbf{N}^{(j-1)}[\mathbf{I}_p - \mathbf{u}_{j-1}\mathbf{p}_{j-1}^{\mathtt{H}}]$, $j = 2, \ldots, r$. Thus, $\mathbf{U}_j$ and $\mathbf{R}_j$ span the same space.

5. The algorithm effects a bilinear decomposition of $\mathbf{X}$ as in (17), where $\mathbf{E}^{(r)} = \mathbf{X}^{(r+1)} \perp \mathbf{T}_r$. Similar comments can be made for $\mathbf{Y}$.

6. QPLS-W2A is based on a rank-one deflation of individual block matrices using the corresponding score and loading vectors in the following way:

$$\mathbf{X}^{(j+1)} = \mathbf{X}^{(j)} - \mathbf{t}_j \mathbf{p}_j^{\mathtt{H}}, \qquad \mathbf{Y}^{(j+1)} = \mathbf{Y}^{(j)} - \mathbf{s}_j \mathbf{q}_j^{\mathtt{H}} \tag{19}$$

QPLS-W2A is summarized in Algorithm 1.

**Algorithm 1** (QPLS-W2A)

**Require:** $\mathbf{X} \in \mathbb{Q}^{n \times p}$ and $\mathbf{Y} \in \mathbb{Q}^{n \times q}$

**Ensure:** $\{\mathbf{t}_j\}_{j=1}^r$, $\{\mathbf{p}_j\}_{j=1}^r$, $\{\mathbf{s}_j\}_{j=1}^r$, and $\{\mathbf{q}_j\}_{j=1}^r$ $(r \leq p)$

1: $\mathbf{X}^{(1)} \leftarrow \mathbf{X}$ and $\mathbf{Y}^{(1)} \leftarrow \mathbf{Y}$

2: **for** $j = 1, \ldots, r$ **do**

3: $\quad (\mathbf{u}_j, \mathbf{v}_j) \leftarrow$ first pair of singular vectors of $\mathbf{X}^{(j)\mathtt{H}} \mathbf{Y}^{(j)}$

4: $\quad \mathbf{t}_j \leftarrow \mathbf{X}^{(j)} \mathbf{u}_j$ and $\mathbf{s}_j \leftarrow \mathbf{Y}^{(j)} \mathbf{v}_j$

5: $\quad \mathbf{p}_j^{\mathtt{H}} \leftarrow \mathbf{t}_j^{+} \mathbf{X}^{(j)}$ and $\mathbf{q}_j^{\mathtt{H}} \leftarrow \mathbf{s}_j^{+} \mathbf{Y}^{(j)}$

6: $\quad \mathbf{X}^{(j+1)} \leftarrow \mathbf{X}^{(j)} - \mathbf{t}_j \mathbf{p}_j^{\mathtt{H}}$ and $\mathbf{Y}^{(j+1)} \leftarrow \mathbf{Y}^{(j)} - \mathbf{s}_j \mathbf{q}_j^{\mathtt{H}}$

7: **end for**

### *3.3. QPLS2 algorithm*

Unlike QPLS-W2A, QPSL2 treats the relationship between $\mathbf{X}$ and $\mathbf{Y}$ in an asymmetrical way [21], i.e., in contrast to the latent decomposition (17) assumed by QPLS-W2A, QPLS2 regression is based on the specific latent decomposition:

$$\begin{aligned} \mathbf{Y} &= \mathbf{T}_r \mathbf{Q}_r^{\mathtt{H}} + \mathbf{F}^{(r)} \\ \mathbf{X} &= \mathbf{T}_r \mathbf{P}_r^{\mathtt{H}} + \mathbf{E}^{(r)} \end{aligned} \tag{20}$$

This asymmetric assumption of the regressor-response variables relation is transformed into a deflaction scheme where the $\mathbf{t}_j$ score vectors are good predictors of $\mathbf{Y}$. Now, the deflaction at each iteration for the $\mathbf{Y}^{(j)}$ block matrices is

$$\mathbf{Y}^{(j+1)} = \mathbf{Y}^{(j)} - \mathbf{t}_j \mathbf{q}_j^{\mathtt{H}} \tag{21}$$

with $\mathbf{q}_j^{\mathtt{H}} = \mathbf{t}_j^{+}\mathbf{Y}^{(j)}$, while the rest remains unchanged.

The score and loading vectors of both algorithms share similar properties (see Theorem 1). QPSL2 is summarized in Algorithm 2.

**Algorithm 2** (QPLS2)

**Require:** $\mathbf{X} \in \mathbb{Q}^{n\times p}$ and $\mathbf{Y} \in \mathbb{Q}^{n\times q}$

**Ensure:** $\{\mathbf{t}_j\}_{j=1}^{r}$, $\{\mathbf{p}_j\}_{j=1}^{r}$, and $\{\mathbf{q}_j\}_{j=1}^{r}$ $(r \leq p)$

1: $\mathbf{X}^{(1)} \leftarrow \mathbf{X}$ and $\mathbf{Y}^{(1)} \leftarrow \mathbf{Y}$

2: **for** $j = 1, \ldots, r$ **do**

3: $\quad \mathbf{u}_j \leftarrow$ first left singular vector of $\mathbf{X}^{(j)\mathtt{H}}\mathbf{Y}^{(j)}$

4: $\quad \mathbf{t}_j \leftarrow \mathbf{X}^{(j)}\mathbf{u}_j$

5: $\quad \mathbf{p}_j^{\mathtt{H}} \leftarrow \mathbf{t}_j^{+}\mathbf{X}^{(j)}$ and $\mathbf{q}_j^{\mathtt{H}} \leftarrow \mathbf{t}_j^{+}\mathbf{Y}^{(j)}$

6: $\quad \mathbf{X}^{(j+1)} \leftarrow \mathbf{X}^{(j)} - \mathbf{t}_j\mathbf{p}_j^{\mathtt{H}}$ and $\mathbf{Y}^{(j+1)} \leftarrow \mathbf{Y}^{(j)} - \mathbf{t}_j\mathbf{q}_j^{\mathtt{H}}$

7: **end for**

### *3.4. QSIMPLS algorithm*

QSIMPLS is another QPLS method that differs from those based on NIPALS in two important ways: first, successive $\mathbf{t}_j$ are calculated explicitly as linear combinations of $\mathbf{X}$; second, $\mathbf{X}$ is not deflated in the same way. These characteristics may result in faster computation and less memory requirements. The real-valued version of this algorithm was suggested by de Jong [22].

The application of this algorithm is based on the following result.

**Theorem 2.** *The following assertion holds:*

1. *Define* $\mathbf{t}_j = \mathbf{X}\mathbf{u}_j$*, where* $\mathbf{u}_j$ *is the first left singular vector of*

$$\mathbf{Z}^{(j)} = (\mathbf{I}_p - \mathbf{P}_{j-1}\mathbf{P}_{j-1}^{+})\mathbf{X}^{\mathtt{H}}\mathbf{Y} \tag{22}$$

   *and* $\mathbf{P}_{j-1} = [\mathbf{p}_1, \ldots, \mathbf{p}_{j-1}]$ *is the matrix of loadings with* $\mathbf{p}_i^{\mathtt{H}} = \mathbf{t}_i^{+}\mathbf{X}$ *then,* $\mathbf{t}_i \perp \mathbf{t}_j$*,* $\forall i < j$*.*

2. *Let* $\mathbf{\Gamma}_j = [\boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_j]$ *be an orthonormal basis of* $\mathbf{P}_j$ *then,*

$$\mathbf{Z}^{(j)} = (\mathbf{I}_p - \boldsymbol{\gamma}_{j-1}\boldsymbol{\gamma}_{j-1}^{\mathtt{H}})\mathbf{Z}^{(j-1)}, \qquad j = 1, \ldots, r$$
$$\mathbf{Z}^{(0)} = \mathbf{X}^{\mathtt{H}}\mathbf{Y}$$

3. $\hat{\mathbf{Y}} = \mathbf{\Pi}_{\mathbf{T}_r}\mathbf{Y} = \mathbf{X}\hat{\mathbf{B}}_{\text{SIM}}$ *with* $\hat{\mathbf{B}}_{\text{SIM}} = \mathbf{U}_r\mathbf{T}_r^{+}\mathbf{Y}$ *and* $\mathbf{U}_r = [\mathbf{u}_1, \ldots, \mathbf{u}_r]$.

Proof. See Appendix A.2.

**Algorithm 3** (QSIMPLS)

**Require:** $\mathbf{X} \in \mathbb{Q}^{n\times p}$ and $\mathbf{Y} \in \mathbb{Q}^{n\times q}$

**Ensure:** $\{\mathbf{t}_j\}_{j=1}^{r}$ and $\{\mathbf{p}_j\}_{j=1}^{r}$ $(r \leq p)$

1: $\mathbf{Z}^{(1)} \leftarrow \mathbf{X}^{\mathtt{H}}\mathbf{Y}$

2: **for** $j = 1, \ldots, r$ **do**

3: $\quad\mathbf{u}_j \leftarrow$ first left singular vector of $\mathbf{Z}^{(j)}$

4: $\quad\mathbf{t}_j \leftarrow \mathbf{X}\mathbf{u}_j$

5: $\quad\mathbf{p}_j^{\mathtt{H}} \leftarrow \mathbf{t}_j^{+}\mathbf{X}$

6: $\quad\boldsymbol{\gamma}_j \leftarrow$ orthonormalize $\mathbf{p}_j$ (Gram-Schmidt technique)

7: $\quad\mathbf{Z}^{(j+1)} \leftarrow (\mathbf{I}_p - \boldsymbol{\gamma}_j\boldsymbol{\gamma}_j^{\mathtt{H}})\mathbf{Z}^{(j)}$

8: **end for**

### *3.5. QPCovR algorithm*

PCovR aims to find components that not only summarize the scores on the predictor variables well, but also predict the scores on the response variables well. To this end, this technique introduces an extra weighting parameter, $\alpha$, which allows us to flexibly choose how much we wish to emphasize reconstruction and prediction [23]. Based on these ideas, we propose a new approach for the quaternion domain. Specifically, assuming latent decomposition (20), the following optimization problem has to be solved:

$$\min_{\mathbf{W}_r} \left\{\alpha\|\mathbf{X} - \mathbf{X}\mathbf{W}_r\mathbf{P}_r^{\mathtt{H}}\|^2 + (1-\alpha)\|\mathbf{Y} - \mathbf{Y}\mathbf{W}_r\mathbf{Q}_r^{\mathtt{H}}\|^2\right\}, \qquad \alpha \in [0,1] \tag{23}$$

subject to $\mathbf{T}_r^{\mathtt{H}}\mathbf{T}_r = \mathbf{I}_r$, where $\mathbf{T}_r = \mathbf{X}\mathbf{W}_r$.

Clearly, when $\alpha = 1$, the technique focuses entirely on summarizing $\mathbf{X}$, which corresponds to QPCR. When $\alpha = 0$, it focuses entirely on predicting $\mathbf{Y}$, which gives us quaternion reduced rank regression, or, when the number of components is at least as large as the number of response variables, OLS

regression. A problem associated with QPCovR is that a choice for $\alpha$ has to be made. As will see later, cross-validation can be used for this purpose.

Two closed-form solutions are available for extracting the components as shown in the following result.

**Theorem 3.** 1. *A solution to the optimization problem* (23) *consists of selecting* $\mathbf{T}_r$ *as the first* $r$ *right eigenvectors of the matrix:*

$$\alpha\mathbf{X}\mathbf{X}^{\mathtt{H}} + (1-\alpha)\hat{\mathbf{Y}}\hat{\mathbf{Y}}^{\mathtt{H}} \tag{24}$$

*with* $\hat{\mathbf{Y}} = \mathbf{X}\mathbf{X}^{+}\mathbf{Y}$. *Also,* $\mathbf{W}_r = \mathbf{X}^{+}\mathbf{T}_r$.

2. *Another solution to the optimization problem* (23) *consists of selecting* $(\mathbf{X}^{\mathtt{H}}\mathbf{X})^{1/2}\mathbf{W}_r$ *as the first* $r$ *right eigenvectors of the matrix:*

$$(\mathbf{X}^{\mathtt{H}}\mathbf{X})^{1/2}\left[\alpha\mathbf{I}_p + (1-\alpha)(\mathbf{X}^{\mathtt{H}}\mathbf{X})^{-1}\mathbf{X}^{\mathtt{H}}\mathbf{Y}\mathbf{Y}^{\mathtt{H}}\mathbf{X}(\mathbf{X}^{\mathtt{H}}\mathbf{X})^{-1}\right](\mathbf{X}^{\mathtt{H}}\mathbf{X})^{1/2} \tag{25}$$

3. $\mathbf{P}_r^{\mathtt{H}} = \mathbf{T}_r^{\mathtt{H}}\mathbf{X}$ *and* $\mathbf{Q}_r^{\mathtt{H}} = \mathbf{T}_r^{\mathtt{H}}\mathbf{Y}$.
4. $\hat{\mathbf{Y}} = \mathbf{\Pi}_{\mathbf{T}_r}\mathbf{Y} = \mathbf{X}\hat{\mathbf{B}}_{\mathrm{PCovR}}$ *with* $\hat{\mathbf{B}}_{\mathrm{PCovR}} = \mathbf{X}^{+}\mathbf{T}_r\mathbf{T}_r^{\mathtt{H}}\mathbf{Y}$.

Proof. See Appendix A.3.

This result provides two different closed-form solutions to the optimization problem (23). In the first one, an eigendecomposition of (24), which is a $n \times n$ matrix, must be derived and thus, $\mathbf{T}_r$ is $n \times r$. In the second one, the matrix (25) is $p \times p$. So, the choice depends on the relative sizes of $n$ and $p$. When $n > p$ the second option is the preferred. Finally, in the first solution, it is obvious that $\mathbf{T}_r^{\mathtt{H}}\mathbf{T}_r = \mathbf{I}_r$, and in the second one, the first $r$ right eigenvectors of (25) are denoted as $\mathbf{U}_r = (\mathbf{X}^{\mathtt{H}}\mathbf{X})^{1/2}\mathbf{W}_r$, then

$$\mathbf{T}_r^{\mathtt{H}}\mathbf{T}_r = \mathbf{W}_r^{\mathtt{H}}\mathbf{X}^{\mathtt{H}}\mathbf{X}\mathbf{W}_r = \mathbf{U}_r^{\mathtt{H}}(\mathbf{X}^{\mathtt{H}}\mathbf{X})^{-1/2}\mathbf{X}^{\mathtt{H}}\mathbf{X}(\mathbf{X}^{\mathtt{H}}\mathbf{X})^{-1/2}\mathbf{U}_r = \mathbf{I}_r \tag{26}$$

### *3.6. Application and comparison of the algorithms*

As highlighted in Subsection 2.3, the adequate processing in quaternion estimation is the WL processing, i.e., we must operate with augmented statistics

to fully capture the second-order statistical properties of the random vectors. As a consequence, the minimum MSE estimate of $\mathbf{Y}$ is expressed as

$$\hat{\mathbf{Y}} = \bar{\mathbf{X}}\hat{\mathbf{B}} = \mathbf{X}\hat{\mathbf{B}}_1 + \mathbf{X}^{\mathtt{i}}\hat{\mathbf{B}}_2 + \mathbf{X}^{\mathtt{j}}\hat{\mathbf{B}}_3 + \mathbf{X}^{\mathtt{k}}\hat{\mathbf{B}}_4 \tag{27}$$

with $\bar{\mathbf{X}} = [\mathbf{X}, \mathbf{X}^{\mathtt{i}}, \mathbf{X}^{\mathtt{j}}, \mathbf{X}^{\mathtt{k}}] \in \mathbb{Q}^{n\times 4p}$ and $\hat{\mathbf{B}} = [\hat{\mathbf{B}}_1^{\mathtt{T}}, \hat{\mathbf{B}}_2^{\mathtt{T}}, \hat{\mathbf{B}}_3^{\mathtt{T}}, \hat{\mathbf{B}}_4^{\mathtt{T}}]^{\mathtt{T}} \in \mathbb{Q}^{4p\times q}$. The above model is referred to as WL quaternion regression. Thus, in general, the information supplied by both $\bar{\mathbf{X}}$ and $\bar{\mathbf{Y}}$ has to be used for calculating $\hat{\mathbf{B}}$.

Within this framework, Stott et al. [8] devised the NIPALS algorithm for WL quaternion PLS (WL-QPLS) which is summarised in Algorithm 4. In its derivation, they make use of the link between the augmented data matrix $\bar{\mathbf{X}}$ and the real-valued matrix $\mathbf{X}_{re} = [\mathbf{X}_0, \mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3] \in \mathbb{R}^{n\times 4p}$, given by the relations

$$\mathbf{X}_{re} = \frac{1}{4}\bar{\mathbf{X}}\boldsymbol{\mathcal{T}}_p^{*}, \qquad \bar{\mathbf{X}} = \mathbf{X}_{re}\boldsymbol{\mathcal{T}}_p^{\mathtt{T}} \tag{28}$$

where $\boldsymbol{\mathcal{T}}_p$ is given in (7).

**Algorithm 4** (WL-QPLS) [8]

**Require:** $\bar{\mathbf{X}} \in \mathbb{Q}^{n\times 4p}$ and $\bar{\mathbf{Y}} \in \mathbb{Q}^{n\times 4q}$

**Ensure:** $\{\mathbf{t}_j\}_{j=1}^r$, $\{\dot{\mathbf{p}}_j\}_{j=1}^r$, $\{\dot{\mathbf{q}}_j\}_{j=1}^r$ and $\{\mathbf{w}_j\}_{j=1}^r$ $(r \le p)$

1: $\bar{\mathbf{X}}^{(1)} \leftarrow \bar{\mathbf{X}}$ and $\bar{\mathbf{Y}}^{(1)} \leftarrow \bar{\mathbf{Y}}$

2: **for** $j = 1, \ldots, r$ **do**

3: $\quad \mathbf{X}_{re}^{(j)} \leftarrow \bar{\mathbf{X}}^{(j)}\boldsymbol{\mathcal{T}}_p$ and $\mathbf{Y}_{re}^{(j)} \leftarrow \bar{\mathbf{Y}}^{(j)}\boldsymbol{\mathcal{T}}_p$

4: $\quad \mathbf{w}_{re}^{(j)} \leftarrow \mathrm{Eig}_{\max}\left\{\mathbf{X}_{re}^{(j)\mathtt{T}}\mathbf{Y}_{re}^{(j)}\mathbf{Y}_{re}^{(j)\mathtt{T}}\mathbf{X}_{re}^{(j)}\right\}$

5: $\quad [\mathbf{w}_j^{\mathtt{T}}, \mathbf{w}_j^{\mathtt{i}\mathtt{T}}, \mathbf{w}_j^{\mathtt{j}\mathtt{T}}, \mathbf{w}_j^{\mathtt{k}\mathtt{T}}]^{\mathtt{T}} \leftarrow \boldsymbol{\mathcal{T}}_p\mathbf{w}_{re}^{(j)}$

6: $\quad \mathbf{t}_j \leftarrow \mathbf{X}^{(j)}\mathbf{w}_j$, $\bar{\mathbf{t}}_j = [\mathbf{t}_j, \mathbf{t}_j^{\mathtt{i}}, \mathbf{t}_j^{\mathtt{j}}, \mathbf{t}_j^{\mathtt{k}}]$

7: $\quad \dot{\mathbf{p}}_j^{\mathtt{H}} \leftarrow \bar{\mathbf{t}}_j^{+}\bar{\mathbf{X}}^{(j)}$ and $\dot{\mathbf{q}}_j^{\mathtt{H}} \leftarrow \bar{\mathbf{t}}_j^{+}\bar{\mathbf{Y}}^{(j)}$

8: $\quad \bar{\mathbf{X}}^{(j+1)} \leftarrow \bar{\mathbf{X}}^{(j)} - \bar{\mathbf{t}}_j\dot{\mathbf{p}}_j^{\mathtt{H}}$ and $\bar{\mathbf{Y}}^{(j+1)} \leftarrow \bar{\mathbf{Y}}^{(j)} - \bar{\mathbf{t}}_j\dot{\mathbf{q}}_j^{\mathtt{H}}$

9: **end for**

WL-QPLS, QPLS-W2A and QPLS2 are all based on NIPALS, but they give different results when applied on $\bar{\mathbf{X}}$ and $\bar{\mathbf{Y}}$. Actually, QPLS2 is most similar to WL-QPLS as both algorithms treat the data matrices in an asymmetrical way,

i.e., both assume a latent model of type (20) for $\bar{\mathbf{X}}$ and $\bar{\mathbf{Y}}$. In particular, WL-QPLS provides the following decompositions for the augmented data matrices

$$\begin{aligned} \bar{\mathbf{Y}} &= \sum_{j=1}^{r} \bar{\mathbf{t}}_j \dot{\mathbf{p}}_j^{\mathtt{H}} + \bar{\mathbf{F}}^{(r)} \\ \bar{\mathbf{X}} &= \sum_{j=1}^{r} \bar{\mathbf{t}}_j \dot{\mathbf{q}}_j^{\mathtt{H}} + \bar{\mathbf{E}}^{(r)} \end{aligned} \tag{29}$$

where a major difference in relation to QPLS2 can be observed: $\bar{\mathbf{t}}_j = [\mathbf{t}_j, \mathbf{t}_j^{\mathtt{i}}, \mathbf{t}_j^{\mathtt{j}}, \mathbf{t}_j^{\mathtt{k}}]$ matrices in (29), derived in step 6 of Algorithm 4, are not orthogonal ($\bar{\mathbf{t}}_i^{\mathtt{H}} \bar{\mathbf{t}}_j$ is block-diagonal). This does not occur in Algorithm 2, as proved in Theorem 1.

Another difference between WL-QPLS and the WL versions of the algorithms suggested here is that the former is not equivalent to a real-valued NIPALS implementation. However, this equivalence is lost when our algorithms are applied under properness conditions since they attain a notable reduction in computational burden when compared to the WL version (see [11] for an exhaustive analysis). This can be understood from the definitions of the properness properties, considering that while in the improper case we should operate with $\bar{\mathbf{X}} \in \mathbb{Q}^{n \times 4p}$ and $\bar{\mathbf{Y}} \in \mathbb{Q}^{n \times 4q}$, in the $\mathbb{C}^{\mathtt{i}}$-proper case we have to manage data matrices of smaller dimensions $\tilde{\mathbf{X}} = [\mathbf{X}, \mathbf{X}^{\mathtt{i}}] \in \mathbb{Q}^{n \times 2p}$ and $\tilde{\mathbf{Y}} = [\mathbf{Y}, \mathbf{Y}^{\mathtt{i}}] \in \mathbb{Q}^{n \times 2q}$, and under $\mathbb{Q}$-properness we work directly with $\mathbf{X} \in \mathbb{Q}^{n \times p}$ and $\mathbf{Y} \in \mathbb{Q}^{n \times q}$ (see (10), (13) and (14)). Furthermore, this reduction in computational burden cannot be achieved in the real domain, making the proper approach more attractive. Thus, the properness of quaternion random vectors should be routinely checked in experimental studies, for which the statistical hypotheses tests suggested in [24], [25] can be used. Finally, real applications where proper quaternion random vectors appear can be found in [26], [27], [28], among others.

## 4. Simulations

Two types of numerical simulations have been designed to assess the performance of the algorithms suggested when compared with the QPLS-SB [10] and WL-QPLS solutions [8]. Their performance has been measured in terms of

the prediction capability of the algorithms and their degree of accuracy in estimating the correct number of components. The first kind of simulation is based on $\mathbf{X}$ data matrices which are nearly singular (some of their singular values are very close to 0). This type of ill-conditioning is less informative than the second one considered, where the $\mathbf{X}$ matrices are generated from a prefixed number $r$ of components. This second option imposes a stronger structure on $\mathbf{X}$ which benefits the performance of the algorithms, as we will see later. Moreover, both kinds of ill-conditioned matrices are analyzed under an impropriety scenario (WL processing) and also in the case of $\mathbb{Q}$-properness (for $\mathbf{X}$ nearly singular) and $\mathbb{C}^{\mathbf{i}}$-properness (for $\mathbf{X}$ with $r$ prefixed components).

All the simulations share some characteristics that are detailed below. By using a Monte Carlo simulation, we have generated $N = 200$ values of the $\bar{\mathbf{X}}$ and $\bar{\mathbf{Y}}$ data matrices in the WL case, $\tilde{\mathbf{X}}$ and $\tilde{\mathbf{Y}}$ in the SWL case, and $\mathbf{X}$ and $\mathbf{Y}$ in the SL scenario. This procedure has been repeated in 100 runs. In each run, the sample was split into two equal parts: $N/2 = 100$ data for the training sample and $N/2 = 100$ for the validation sample. The training samples allow us to estimate the regression coefficients matrix: $\hat{\mathbf{B}}_{j,r}$, $j = 1, \ldots, 100$ (runs); $r = 1, \ldots, R$, where $R$ is the maximum number of components analyzed ($R = p$ in the SL case, $R = 2p$ in the SWL case and $R = 4p$ in the WL case). The validation samples are used for calculating the following prediction performance measure

$$MSE_{j,r} = \frac{\|\mathbf{Y}_j - \hat{\mathbf{Y}}_{j,r}\|}{100 - r + 1}, \qquad j = 1, \ldots, 100; \quad r = 1, \ldots, R \tag{30}$$

where the term $r$ in the denominator penalizes the inclusion of additional components [29]. Then, we obtain

$$\overline{MSE}_r = \sum_{j=1}^{100} \frac{MSE_{j,r}}{100}, \qquad r = 1, \ldots, R \tag{31}$$

and select the optimum number of components as $\min_r \overline{MSE}_r$. The simulation procedure is also used to estimate the parameter $\alpha$ needed for QPCovR. Specifically, the above procedure is repeated for QPCovR on a mesh of values

of $\alpha \in [0, 1]$. So, the prediction error measure in (31) depends on an additional argument for this algorithm: $\overline{MSE}_r(\alpha)$. Now, the optimum $\alpha$ and the optimum number of components are selected as $\min_{r,\alpha} \overline{MSE}_r(\alpha)$.

The simulations are also performed under three levels of noise: low ($\sigma^2 = 0.01$), medium ($\sigma^2 = 1$), and high ($\sigma^2 = 100$), i.e., $\mathbf{\Omega}$ in (15) is a noise matrix generated from a $q$-dimensional Gaussian quaternion white noise $\boldsymbol{\omega}$ with $\mathrm{E}[\boldsymbol{\omega}\boldsymbol{\omega}^{\mathrm{H}}] = 4\sigma^2\mathbf{I}_q$. Finally, different $\mathbf{B}$ matrices were randomly generated in each case analyzed.

### *4.1. Case 1: WL scenario and* $\mathbf{X}$ *nearly singular*

A data matrix $\bar{\mathbf{X}} \in \mathbb{Q}^{200\times 40}$ was generated from an improper quaternion Gaussian vector with a nearly singular covariance matrix. The response data matrix, $\mathbf{Y} \in \mathbb{Q}^{200\times 10}$, was then calculated as

$$\mathbf{Y} = \bar{\mathbf{X}}\mathbf{B} + \mathbf{\Omega} \tag{32}$$

Next, the algorithms are applied on $\bar{\mathbf{X}}$ and $\bar{\mathbf{Y}}$ and the results are depicted in Table 1. Since $p = 10$, then the maximum number of components to analyze is $R = 4p = 40$. Moreover, the optimum values of $\alpha$ for the QPCovR algorithm appear in parentheses.

Table 1 shows that a lack of a defined structure in the ill-conditioning of data matrix $\mathbf{X}$ leads to a diversity of results. The most important characteristics observed shows that, in general, the optimum number of components decreases as the noise level increases, i.e., all the methods are seriously affected by high noise levels. Nevetheless, QPCovR always shows the better performance: it needs the lowest number of components to attain the smallest $\overline{MSE}_r$.

### *4.2. Case 2: SL scenario and* $\mathbf{X}$ *nearly singular*

In this case, matrix $\mathbf{X} \in \mathbb{Q}^{200\times 20}$ was generated from a $\mathbb{Q}$-proper random vector distributed as a multivariate Gaussian distribution with a nearly singular covariance matrix. The matrix $\mathbf{Y} \in \mathbb{Q}^{200\times 20}$ was obtained as $\mathbf{Y} = \mathbf{X}\mathbf{B} + \mathbf{\Omega}$. The algorithms are now applied directly on $\mathbf{X}$ and $\mathbf{Y}$, and the results are shown in Table 2.

| Case 1 | | | | | | |
|---|---|---|---|---|---|---|
| $\sigma^2 = 0.01 \qquad (\alpha = 0.7)$ | | | | | | |
| | QPLS-SB | QPLS-W2A | QPLS2 | QSIMPLS | QPCovR | WL-QPLS |
| $r$ | 35 | 35 | 31 | 31 | **23** | 36 |
| $\overline{MSE}_r$ | 0.124 | 0.124 | 0.115 | 0.115 | **0.104** | 0.125 |
| $\sigma^2 = 1 \qquad (\alpha = 0.1)$ | | | | | | |
| | QPLS-SB | QPLS-W2A | QPLS2 | QSIMPLS | QPCovR | WL-QPLS |
| $r$ | 26 | 26 | 22 | 22 | **15** | 28 |
| $\overline{MSE}_r$ | 1.018 | 1.019 | 0.952 | 0.954 | **0.892** | 1.048 |
| $\sigma^2 = 100 \qquad (\alpha = 0.2)$ | | | | | | |
| | QPLS-SB | QPLS-W2A | QPLS2 | QSIMPLS | QPCovR | WL-QPLS |
| $r$ | 17 | 17 | 13 | 13 | **6** | 16 |
| $\overline{MSE}_r$ | 8.434 | 8.428 | 8.015 | 8.043 | **7.541** | 8.573 |

Table 1: $\overline{MSE}_r$ and the optimum number of components ($r$) for the WL regression algorithms.

The conclusions parallel those of Case 1. That is, the noise level severely affects the results and although there are very few differences between the algorithms, QPCovR always outperforms all the other algorithms in $\overline{MSE}_r$ and this is also achieved with the smallest number of components. As a consequence, once again, QPCovR provides the more parsimonious accurate models for prediction. The similar performance results between QPLS2 and QSIMPLS in both Cases 1 and 2 are also notable.

### *4.3. Case 3: SWL scenario and* $\mathbf{X}$ *with a prefixed number* $r$ *of components*

The experiment is designed in such a way that the data observed are generated by a system driven by a known number of components. Specifically, the data matrix $\mathbf{X} \in \mathbb{Q}^{200\times 20}$ was generated from a $\mathbb{C}^{\mathbf{i}}$-proper random vector whose covariance matrix is forced to have $r = 7$ components. Matrix $\mathbf{Y} \in \mathbb{Q}^{200\times 20}$ was obtained as $\mathbf{Y} = \tilde{\mathbf{X}}\mathbf{B} + \mathbf{\Omega}$. The algorithms were applied on $\tilde{\mathbf{X}}$ and $\tilde{\mathbf{Y}}$ and the results are presented in Table 3.

| Case 2 | | | | | | |
|---|---|---|---|---|---|---|
| $\sigma^2 = 0.01 \quad (\alpha = 0.8)$ | | | | | | |
| | QPLS-SB | QPLS-W2A | QPLS2 | QSIMPLS | QPCovR | WL-QPLS |
| $r$ | 9 | 9 | 9 | 9 | **7** | 9 |
| $\overline{MSE}_r$ | 0.102 | 0.102 | 0.102 | 0.102 | **0.101** | 0.102 |
| $\sigma^2 = 1 \quad (\alpha = 0.8)$ | | | | | | |
| | QPLS-SB | QPLS-W2A | QPLS2 | QSIMPLS | QPCovR | WL-QPLS |
| $r$ | 7 | 7 | 6 | 6 | **4** | 7 |
| $\overline{MSE}_r$ | 0.984 | 0.984 | 0.982 | 0.982 | **0.975** | 0.984 |
| $\sigma^2 = 100 \quad (\alpha = 0.9)$ | | | | | | |
| | QPLS-SB | QPLS-W2A | QPLS2 | QSIMPLS | QPCovR | WL-QPLS |
| $r$ | 4 | 4 | 4 | 4 | **2** | 4 |
| $\overline{MSE}_r$ | 14.009 | 14.008 | 13.986 | 13.986 | **13.714** | 13.999 |

Table 2: $\overline{MSE}_r$ and the optimum number of components ($r$) for the SL regression algorithms.

As we can see, the conclusions are completely different from the previous cases. The induced structure over $\mathbf{X}$ is clearly shown by all the algorithms for any noise levels. Actually, all of them coincide in identifying the correct number of components and in prediction performance. In this situation, QPCovR is the less advantageous option because of the extra computational effort needed for the estimation of the optimum $\alpha$.

### *4.4. Case 4: WL scenario and* $\mathbf{X}$ *with a prefixed number* $r$ *of components*

At this point, we force $\mathbf{X}$ to have $r = 5$ components and then, $\bar{\mathbf{X}}$ and $\bar{\mathbf{Y}}$ are generated from (32), as in Case 1. The results are illustrated in Table 4 for the three different values of noise variance. We can observe that the algorithms perform in a manner that is similar to Case 3. Interestingly, in both Cases 3 and 4, and under high noise levels, the estimation of weighting parameter $\alpha$ is equal to 1, i.e., QPCovR boils down to QPCR.

| Case 3 | | | | | | |
|---|---|---|---|---|---|---|
| $\sigma^2 = 0.01 \qquad (\alpha = 0.4)$ | | | | | | |
| | QPLS-SB | QPLS-W2A | QPLS2 | QSIMPLS | QPCovR | WL-QPLS |
| $r$ | 7 | 7 | 7 | 7 | 7 | 7 |
| $\overline{MSE}_r$ | 0.098 | 0.098 | 0.098 | 0.098 | 0.098 | 0.098 |
| $\sigma^2 = 1 \qquad (\alpha = 0.2)$ | | | | | | |
| | QPLS-SB | QPLS-W2A | QPLS2 | QSIMPLS | QPCovR | WL-QPLS |
| $r$ | 7 | 7 | 7 | 7 | 7 | 7 |
| $\overline{MSE}_r$ | 0.982 | 0.982 | 0.982 | 0.982 | 0.982 | 0.982 |
| $\sigma^2 = 100 \qquad (\alpha = 1)$ | | | | | | |
| | QPLS-SB | QPLS-W2A | QPLS2 | QSIMPLS | QPCovR | WL-QPLS |
| $r$ | 7 | 7 | 7 | 7 | 7 | 7 |
| $\overline{MSE}_r$ | 9.829 | 9.829 | 9.829 | 9.829 | 9.829 | 9.829 |

Table 3: $\overline{MSE}_r$ and the optimum number of components ($r$) for the SWL regression algorithms.

### *4.5. Discussion and recommendations*

Based on the simulations presented here, some conclusions can be drawn, and some recommendations can be given. First of all, predictability is always quite good in low noise conditions (with all methods and any degrees of properness). For medium and high noise levels, predictability is poorer than for low noise levels.

The results also show that the decomposition of an ill-conditioned data matrix does not necessarily have a clearly determined number of components. In fact, as illustrated in Cases 1 and 2, different algorithms propose different solutions. This issue depends on noise levels and appears with both improper and proper random vectors. In this setting, it is thus advisable to have a set of BRMs available, and to select the best one on the basis of the particular data at hand. Among them, QPCovR has shown good performance in all the scenarios analyzed, followed by QSIMPLS and QPLS2.

| Case 4 | | | | | | |
|---|---|---|---|---|---|---|
| $\sigma^2 = 0.01 \qquad (\alpha = 0.8)$ | | | | | | |
| | QPLS-SB | QPLS-W2A | QPLS2 | QSIMPLS | QPCovR | WL-QPLS |
| $r$ | 5 | 5 | 5 | 5 | 5 | 5 |
| $\overline{MSE}_r$ | 0.067 | 0.067 | 0.067 | 0.067 | 0.067 | 0.067 |
| $\sigma^2 = 1 \qquad (\alpha = 0.7)$ | | | | | | |
| | QPLS-SB | QPLS-W2A | QPLS2 | QSIMPLS | QPCovR | WL-QPLS |
| $r$ | 5 | 5 | 5 | 5 | 5 | 5 |
| $\overline{MSE}_r$ | 0.672 | 0.672 | 0.672 | 0.672 | 0.672 | 0.672 |
| $\sigma^2 = 100 \qquad (\alpha = 1)$ | | | | | | |
| | QPLS-SB | QPLS-W2A | QPLS2 | QSIMPLS | QPCovR | WL-QPLS |
| $r$ | 5 | 5 | 5 | 5 | 5 | 5 |
| $\overline{MSE}_r$ | 6.724 | 6.724 | 6.724 | 6.724 | 6.724 | 6.724 |

Table 4: $\overline{MSE}_r$ and the optimum number of components ($r$) for the WL regression algorithms.

However, the choice of one specific algorithm is less important when a data matrix exhibits a structure determined by a specific number of components (Cases 3 y 4). Nevertheless, as noted above, the downside of QPCovR is that the parameter $\alpha$ has to be tuned, QPLS-SB always provides non-orthogonal components and WL-QPLS also gives non-orthogonal components whenever a decomposition for $\bar{\mathbf{X}}$, as in (29), (or for $\tilde{\mathbf{X}}$) is required. Thus, QPLS2, QSIMPLS, and QPLS-W2A are recommended in these cases.

It was also striking that in all cases QPLS2 and QSIMPLS both performed almost equally for the kinds of data simulated here. Overall, it can be concluded that QPCovR, QSIMPLS, and QPLS2 are the most reliable methods.

## 5. Application of the algorithms in image classification

The algorithms provided were implemented for the real-world problem of color face recognition. To this end, we employed the QLRC approach suggested

in [6] and we compared all the algorithms analyzed in Section 4 with the QLRC solution given in the above paper. We aimed to evaluate their performances on a benchmark color face database: the FEI database [30], [6]. This database contains 2800 images of 200 subjects taken against a homogenous white background. In this experiment, a sample of 50 subjects was selected and 13 images per subject were considered: 10 images for training and 3 for testing. All images were down-sampled to size $48 \times 64$ as in [6].

Figure 1 depicts the condition numbers of the matrices $\mathbf{X}_l^{\mathtt{H}}\mathbf{X}_l$, $l = 1, \ldots, 50$, associated with each QLR model to be estimated (one per subject). The minimum condition number is 2.0337e+03 and the maximum one is 3.6688e+04, and thus, they are all ill-conditioned matrices. Table 5 shows the recognition accuracy of all the methods under comparison, and we selected $r = 5$ components in all the biased regression algorithms, and we considered QPCovR on a mesh of $\alpha = 0, 0.1, 0.2, \ldots, 1$ values. The highest performances were achieved by QPLS2, QSIMPLS, QPCovR ($\alpha = 0.3$), and WL-QPLS. Although the results of the four methods above were identical, QSIMPLS was the most computationally efficient, as shown in Table 6. In conclusion, QSIMPLS is the appropriate choice in this example.

| QLRC | QPLS-SB | QPLS-W2A | QPLS2 | QSIMPLS | QPCovR | WL-QPLS |
|---|---|---|---|---|---|---|
| 93% | 89% | 92% | **95%** | **95%** | **95%** | **95%** |

Table 5: Comparison in accuracy using all methods.

| QLRC | QPLS-SB | QPLS-W2A | QPLS2 | QSIMPLS | QPCovR | WL-QPLS |
|---|---|---|---|---|---|---|
| **5.22 s** | 5.26 s | 60.59 s | 33.08 s | 5.41 s | 58.64 s | 56.26 s |

Table 6: Average recognition CPU time over 10 runs in each method. All algorithms are implemented on an Intel Core i7 2.6GHz CPU with 32GB RAM computer using Matlab R2019a.

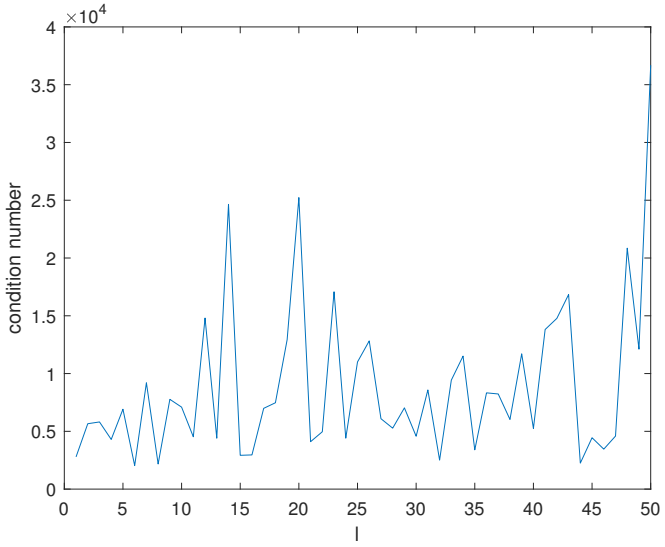


Figure 1: Condition numbers of $\mathbf{X}_l^{\mathrm{H}}\mathbf{X}_l$, $l = 1, \ldots, 50$.

## 6. Conclusions

BRMs allow the problems associated with running OLS regression in ill-conditioned linear models to be dealt with. Despite its benefits, the topic has not been sufficiently developed in the quaternion domain. This has motivated us to propose four new quaternion methods based on dimension-reduction techniques. We have also compared them with other existing solutions through a series of experimental studies. Recommendations for applying the methods in different settings have been made. The most notable conclusions that can be drawn from the current study are: no method outperforms the others in every situation and condition, QPCovR, QSIMPLS and QPLS2 are good choices (QSIMPLS is the fastest one), and the technique that is appropriate for each situation will depend on the data available.

## 7. Acknowledgments

This publication is part of the I+D+i project PID2021-124486NB-I00, funded by MCIN/AEI/10.13039/501100011033/ and by "ERDF A way to make Eu-

rope", and Project EI-FQM2-2023 of "Plan de Apoyo a la Investigación 2023-2024" from the University of Jaén.

## Appendix A. Proofs for the Theorems

### *Appendix A.1. Proof for Theorem 1*

We prove the assertions for the vectors and matrices derived from $\mathbf{X}$. Similar proofs can be carried out for the vectors and matrices obtained from $\mathbf{Y}$.

1. It follows that

$$\mathbf{X}^{(j+1)} = [\mathbf{I}_n - \mathbf{t}_j \mathbf{t}_j^+]\mathbf{X}^{(j)} = \mathbf{A}^{(j)}\mathbf{X}^{(j)} \tag{A.1}$$

with $\mathbf{A}^{(j)} = \mathbf{I}_n - \mathbf{t}_j \mathbf{t}_j^+$. Thus, for $k > j$, we have

$$\mathbf{X}^{(k)} = \mathbf{A}^{(k-1)}\mathbf{X}^{(k-1)} = \mathbf{A}^{(k-1)}\mathbf{A}^{(k-2)} \dots \mathbf{A}^{(j+1)}\mathbf{A}^{(j)}\mathbf{X}^{(j)} = \mathbf{\Theta}\mathbf{A}^{(j)}\mathbf{X}^{(j)} \tag{A.2}$$

where $\mathbf{\Theta} = \mathbf{A}^{(k-1)}\mathbf{A}^{(k-2)} \dots \mathbf{A}^{(j+1)}$. Hence, since $\mathbf{A}^{(j)}\mathbf{t}_j = \mathbf{0}$, we get

$$\mathbf{X}^{(k)}\mathbf{u}_j = \mathbf{\Theta}\mathbf{A}^{(j)}\mathbf{X}^{(j)}\mathbf{u}_j = \mathbf{\Theta}\mathbf{A}^{(j)}\mathbf{t}_j = \mathbf{0} \tag{A.3}$$

On the other hand, $\mathbf{u}_k$ is a right autovector of $\mathbf{X}^{(k)\mathtt{H}}\mathbf{Y}^{(k)}\mathbf{Y}^{(k)\mathtt{H}}\mathbf{X}^{(k)}$ and thus,

$$\mathbf{X}^{(k)\mathtt{H}}\mathbf{Y}^{(k)}\mathbf{Y}^{(k)\mathtt{H}}\mathbf{X}^{(k)}\mathbf{u}_k = \mathbf{u}_k\lambda_k \tag{A.4}$$

where $\lambda_k$ is the associated right eigenvalue. By considering (A.3), it follows that

$$\mathbf{u}_j^{\mathtt{H}}\mathbf{u}_k\lambda_k = \mathbf{u}_j^{\mathtt{H}}\mathbf{X}^{(k)\mathtt{H}}\mathbf{Y}^{(k)}\mathbf{Y}^{(k)\mathtt{H}}\mathbf{X}^{(k)}\mathbf{u}_k = \mathbf{0} \tag{A.5}$$

2. For $k > j$, we have

$$\mathbf{X}^{(k)} = \mathbf{X}^{(k-1)}[\mathbf{I}_n - \mathbf{u}_{k-1}\mathbf{t}_{k-1}^+\mathbf{X}^{(k-1)}] = \mathbf{X}^{(k-1)}\mathbf{B}^{(k-1)} = \mathbf{X}^{(j+1)}\mathbf{\Xi} \tag{A.6}$$

where $\mathbf{B}^{(k-1)} = \mathbf{I}_n - \mathbf{u}_{k-1}\mathbf{t}_{k-1}^+\mathbf{X}^{(k-1)}$ and $\mathbf{\Xi} = \mathbf{B}^{(j+1)}\mathbf{B}^{(j)} \dots \mathbf{B}^{(k-1)}$, and thus, from (A.1), it follows that

$$\begin{aligned}\mathbf{t}_j^{\mathtt{H}}\mathbf{X}^{(k)} = \mathbf{t}_j^{\mathtt{H}}\mathbf{X}^{(j+1)}\mathbf{\Xi} &= \mathbf{t}_j^{\mathtt{H}}\mathbf{A}^{(j)}\mathbf{X}^{(j)}\mathbf{\Xi} \\ &= \mathbf{t}_j^{\mathtt{H}}[\mathbf{I}_n - \mathbf{t}_j\mathbf{t}_j^+]\mathbf{X}^{(j)}\mathbf{\Xi} = (\mathbf{t}_j^{\mathtt{H}} - \mathbf{t}_j^{\mathtt{H}})\mathbf{X}^{(j)}\mathbf{\Xi} = \mathbf{0}\end{aligned}$$

Hence, $\mathbf{t}_j^{\mathtt{H}}\mathbf{t}_k = \mathbf{t}_j^{\mathtt{H}}\mathbf{X}^{(k)}\mathbf{u}_k = \mathbf{0}$.

3. Firstly, we prove that $\mathbf{T}_j = \mathbf{X}\mathbf{R}_j = \mathbf{X}[\mathbf{r}_1, \ldots, \mathbf{r}_j]$, with $\mathbf{r}_1 = \mathbf{u}_1$ and $\mathbf{r}_j = \left[\prod_{i=1}^{j-1}(\mathbf{I}_p - \mathbf{u}_j\mathbf{p}_j^{\mathtt{H}})\right]\mathbf{u}_j$, $j = 1\ldots, r$. Obviously, $\mathbf{t}_1 = \mathbf{X}\mathbf{u}_1 = \mathbf{X}\mathbf{r}_1$. Furthermore, it follows that:

$$\mathbf{t}_2 = \mathbf{X}^{(2)}\mathbf{u}_2 = \mathbf{X}(\mathbf{I}_p - \mathbf{u}_1\mathbf{p}_1^{\mathtt{H}})\mathbf{u}_2 = \mathbf{X}\mathbf{r}_2$$
$$\mathbf{t}_3 = \mathbf{X}^{(3)}\mathbf{u}_3 = \mathbf{X}\Big(\mathbf{I}_p - \mathbf{u}_1\mathbf{p}_1^{\mathtt{H}} - (\mathbf{I}_p - \mathbf{u}_1\mathbf{p}_1^{\mathtt{H}})\mathbf{u}_2\mathbf{p}_2^{\mathtt{H}}\Big)\mathbf{u}_3 = \mathbf{X}\mathbf{r}_3$$

and so on, successively. Now, considering that $\mathbf{t}_j^{+\mathtt{H}} = \mathbf{t}_j(\mathbf{t}_j^{\mathtt{H}}\mathbf{t}_j)^{-1}$ and $\mathbf{t}_i \perp \mathbf{t}_j$,
520 we have

$$\mathbf{p}_j = \mathbf{X}^{(j)\mathtt{H}}\mathbf{t}_j^{+\mathtt{H}} = \Big[\mathbf{X}^{\mathtt{H}} - \sum_{i=1}^{j-1}\mathbf{p}_i\mathbf{t}_i^{\mathtt{H}}\Big]\mathbf{t}_j^{+\mathtt{H}} = \mathbf{X}^{\mathtt{H}}\mathbf{t}_j^{+\mathtt{H}} \tag{A.7}$$

and then, $\mathbf{P}_j = \mathbf{X}^{\mathtt{H}}\mathbf{T}_j^{+\mathtt{H}}$. Hence, $\mathbf{R}_j^{\mathtt{H}}\mathbf{P}_j = \mathbf{I}_j$ and $\mathbf{R}_j = \mathbf{U}_j(\mathbf{P}_j^{\mathtt{H}}\mathbf{U}_j)^{-1}$, with $\mathbf{U}_j = [\mathbf{u}_1, \ldots, \mathbf{u}_j]$.

4. $\mathbf{X}^{(j+1)} = \mathbf{X} - \mathbf{T}_j\mathbf{P}_j^{\mathtt{H}} = [\mathbf{I}_n - \mathbf{T}_j\mathbf{T}_j^{+}]\mathbf{X} = \mathbf{\Pi}_{\mathbf{T}_j^{\perp}}\mathbf{X}$.
5. The predictions of $\mathbf{Y}$ are obtained as

$$\hat{\mathbf{Y}} = \mathbf{\Pi}_{\mathbf{T}_r}\mathbf{Y} = \mathbf{T}_r(\mathbf{T}_r^{\mathtt{H}}\mathbf{T}_r)^{-1}\mathbf{T}_r^{\mathtt{H}}\mathbf{Y} = \mathbf{X}\mathbf{R}_r\mathbf{T}_r^{+}\mathbf{Y} \tag{A.8}$$

525 and taking into account that $\mathbf{R}_r = \mathbf{U}_r(\mathbf{P}_r^{\mathtt{H}}\mathbf{U}_r)^{-1}$, the result follows.

*Appendix A.2. Proof for Theorem 2*

1. Since $\mathbf{u}_j$ is a right eigenvector of $\mathbf{Z}^{(j)}\mathbf{Z}^{(j)\mathtt{H}}$, then $\mathbf{u}_j \in \mathcal{C}(\mathbf{Z}^{(j)})$. Also, $\mathbf{Z}^{(j)} = \mathbf{\Pi}_{\mathbf{P}_{j-1}^{\perp}}\mathbf{X}^{\mathtt{H}}\mathbf{Y}$. Hence, $\mathbf{u}_j \perp \mathbf{p}_i$ for $i < j$. Finally, taking into account that $\mathbf{t}_i^{\mathtt{H}}\mathbf{t}_i\mathbf{p}_i^{\mathtt{H}} = \mathbf{t}_i^{\mathtt{H}}\mathbf{X}$, we have

$$\mathbf{t}_i^{\mathtt{H}}\mathbf{t}_j = \mathbf{t}_i^{\mathtt{H}}\mathbf{X}\mathbf{u}_j = \mathbf{t}_i^{\mathtt{H}}\mathbf{t}_i\mathbf{p}_i^{\mathtt{H}}\mathbf{u}_j = 0 \qquad i < j \tag{A.9}$$

530 2. Since $\mathbf{\Gamma}_j$ is an orthonormal basis, then

$$\mathbf{\Pi}_{\mathbf{\Gamma}_{j-1}^{\perp}} = \mathbf{I}_p - \mathbf{\Gamma}_{j-1}\mathbf{\Gamma}_{j-1}^{\mathtt{H}} = \prod_{i=1}^{j-1}(\mathbf{I}_p - \boldsymbol{\gamma}_{j-1}\boldsymbol{\gamma}_{j-1}^{\mathtt{H}}) \tag{A.10}$$

and thus,

$$\mathbf{Z}^{(j)} = \mathbf{\Pi}_{\mathbf{\Gamma}_{j-1}^{\perp}}\mathbf{Z}^{(0)} = \prod_{i=1}^{j-1}(\mathbf{I}_p - \boldsymbol{\gamma}_{j-1}\boldsymbol{\gamma}_{j-1}^{\mathtt{H}})\mathbf{Z}^{(0)} = (\mathbf{I}_p - \boldsymbol{\gamma}_{j-1}\boldsymbol{\gamma}_{j-1}^{\mathtt{H}})\mathbf{Z}^{(j-1)} \tag{A.11}$$

3. $\hat{\mathbf{Y}} = \mathbf{\Pi}_{\mathbf{T}_r}\mathbf{Y} = \mathbf{T}_r\mathbf{T}_r^{+}\mathbf{Y} = \mathbf{X}\mathbf{U}_r\mathbf{T}_r^{+}\mathbf{Y}$.

*Appendix A.3. Proof for Theorem 3*

1. From Property 1-7, minimizing the criterion function (23) is equivalent to the optimization problem: $\min_{\mathbf{\Phi}(\mathbf{W}_r)} \mathbf{\Psi}(\mathbf{W}_r)$, $\alpha \in [0,1]$, with

$$\begin{aligned}\mathbf{\Psi}(\mathbf{W}_r) = \alpha\|\mathbf{\Phi}(\mathbf{X}) - \mathbf{\Phi}(\mathbf{X})\mathbf{\Phi}(\mathbf{W}_r)\mathbf{\Phi}^{\mathsf{T}}(\mathbf{P}_r)\|^2 \\ + (1-\alpha)\|\mathbf{\Phi}(\mathbf{Y}) - \mathbf{\Phi}(\mathbf{Y})\mathbf{\Phi}(\mathbf{W}_r)\mathbf{\Phi}^{\mathsf{T}}(\mathbf{Q}_r)\|^2\end{aligned}$$

subject to $\mathbf{\Phi}^{\mathsf{T}}(\mathbf{T}_r)\mathbf{\Phi}(\mathbf{T}_r) = \mathbf{I}_{4r}$, where $\mathbf{\Phi}(\cdot)$ is the real adjoint matrix (6). Consider the projections

$$\mathbf{\Pi}_{\mathbf{\Phi}(\mathbf{T}_r)}\mathbf{\Phi}(\mathbf{X}) = \mathbf{\Phi}(\mathbf{T}_r)\mathbf{\Phi}^{\mathsf{T}}(\mathbf{T}_r)\mathbf{\Phi}(\mathbf{X}) = \mathbf{\Phi}(\mathbf{T}_r)\mathbf{\Phi}^{\mathsf{T}}(\mathbf{P}_r) \tag{A.12}$$

and $\mathbf{\Pi}_{\mathbf{\Phi}(\mathbf{T}_r)}\mathbf{\Phi}(\mathbf{Y}) = \mathbf{\Phi}(\mathbf{T}_r)\mathbf{\Phi}^{\mathsf{T}}(\mathbf{Q}_r)$, then

$$\begin{aligned}\mathbf{\Psi}(\mathbf{W}_r) =& \alpha\big\|\big(\boldsymbol{I}_{4r} - \mathbf{\Phi}(\mathbf{T}_r)\mathbf{\Phi}^{\mathsf{T}}(\mathbf{T}_r)\big)\mathbf{\Phi}(\mathbf{X})\big\|^2 \\ &+ (1-\alpha)\big\|\big(\boldsymbol{I}_{4r} - \mathbf{\Phi}(\mathbf{T}_r)\mathbf{\Phi}^{\mathsf{T}}(\mathbf{T}_r)\big)\mathbf{\Phi}(\mathbf{Y})\big\|^2 \\ =& \alpha\,\mathrm{Tr}\big\{\mathbf{\Phi}^{\mathsf{T}}(\mathbf{X})\big(\boldsymbol{I}_{4r} - \mathbf{\Phi}(\mathbf{T}_r)\mathbf{\Phi}^{\mathsf{T}}(\mathbf{T}_r)\big)\mathbf{\Phi}(\mathbf{X})\big\} \\ &+ (1-\alpha)\,\mathrm{Tr}\big\{\mathbf{\Phi}^{\mathsf{T}}(\mathbf{Y})\big(\boldsymbol{I}_{4r} - \mathbf{\Phi}(\mathbf{T}_r)\mathbf{\Phi}^{\mathsf{T}}(\mathbf{T}_r)\big)\mathbf{\Phi}(\mathbf{Y})\big\}\end{aligned}$$

Hence, minimizing $\mathbf{\Psi}(\mathbf{W}_r)$ is equivalent to maximizing

$$\begin{aligned}\mathbf{\Upsilon}(\mathbf{W}_r) =& \alpha\,\mathrm{Tr}\big\{\mathbf{\Phi}^{\mathsf{T}}(\mathbf{X})\mathbf{\Phi}(\mathbf{T}_r)\mathbf{\Phi}^{\mathsf{T}}(\mathbf{T}_r)\mathbf{\Phi}(\mathbf{X})\big\} \\ &+ (1-\alpha)\,\mathrm{Tr}\big\{\mathbf{\Phi}^{\mathsf{T}}(\mathbf{Y})\mathbf{\Phi}(\mathbf{T}_r)\mathbf{\Phi}^{\mathsf{T}}(\mathbf{T}_r)\mathbf{\Phi}(\mathbf{Y})\big\} \\ =& \alpha\,\mathrm{Tr}\big\{\mathbf{\Phi}(\mathbf{T}_r)\mathbf{\Phi}^{\mathsf{T}}(\mathbf{T}_r)\mathbf{\Phi}(\mathbf{X})\mathbf{\Phi}^{\mathsf{T}}(\mathbf{X})\big\} \\ &+ (1-\alpha)\,\mathrm{Tr}\big\{\mathbf{\Phi}(\mathbf{T}_r)\mathbf{\Phi}^{\mathsf{T}}(\mathbf{T}_r)\mathbf{\Phi}(\mathbf{Y})\mathbf{\Phi}^{\mathsf{T}}(\mathbf{Y})\big\} \\ =& \alpha\,\mathrm{Tr}\big\{\mathbf{\Phi}(\mathbf{T}_r)\mathbf{\Phi}^{\mathsf{T}}(\mathbf{T}_r)\mathbf{\Phi}(\mathbf{X})\mathbf{\Phi}^{\mathsf{T}}(\mathbf{X})\big\} \\ &+ (1-\alpha)\,\mathrm{Tr}\big\{\mathbf{\Phi}(\mathbf{T}_r)\mathbf{\Phi}^{\mathsf{T}}(\mathbf{T}_r)\mathbf{\Phi}(\mathbf{Y})\mathbf{\Phi}^{\mathsf{T}}(\mathbf{Y})\mathbf{\Phi}(\mathbf{T}_r)\mathbf{\Phi}^{\mathsf{T}}(\mathbf{T}_r)\big\}\end{aligned}$$

Now, consider the projection

$$\hat{\mathbf{\Phi}}(\mathbf{Y}) = \mathbf{\Pi}_{\mathbf{\Phi}(\mathbf{X})}\mathbf{\Phi}(\mathbf{Y}) = \mathbf{\Phi}(\mathbf{X})\mathbf{\Phi}^{+}(\mathbf{X})\mathbf{\Phi}(\mathbf{Y}) \tag{A.13}$$

Since the matrix $\mathbf{\Phi}(\mathbf{T}_r)\mathbf{\Phi}^{\mathsf{T}}(\mathbf{T}_r)$ projects onto a subspace of $\mathcal{C}(\mathbf{\Phi}(\mathbf{X}))$, then

$\boldsymbol{\Phi}(\mathbf{T}_r)\boldsymbol{\Phi}^{\mathtt{T}}(\mathbf{T}_r)\boldsymbol{\Phi}(\mathbf{Y}) = \boldsymbol{\Phi}(\mathbf{T}_r)\boldsymbol{\Phi}^{\mathtt{T}}(\mathbf{T}_r)\hat{\boldsymbol{\Phi}}(\mathbf{Y})$ and thus,

$$\begin{aligned}\boldsymbol{\Upsilon}(\mathbf{W}_r) =& \alpha \operatorname{Tr}\left\{\boldsymbol{\Phi}(\mathbf{T}_r)\boldsymbol{\Phi}^{\mathtt{T}}(\mathbf{T}_r)\boldsymbol{\Phi}(\mathbf{X})\boldsymbol{\Phi}^{\mathtt{T}}(\mathbf{X})\right\} \\ &+ (1-\alpha)\operatorname{Tr}\left\{\boldsymbol{\Phi}(\mathbf{T}_r)\boldsymbol{\Phi}^{\mathtt{T}}(\mathbf{T}_r)\hat{\boldsymbol{\Phi}}(\mathbf{Y})\hat{\boldsymbol{\Phi}}^{\mathtt{T}}(\mathbf{Y})\boldsymbol{\Phi}(\mathbf{T}_r)\boldsymbol{\Phi}^{\mathtt{T}}(\mathbf{T}_r)\right\} \\ =& \alpha \operatorname{Tr}\left\{\boldsymbol{\Phi}(\mathbf{T}_r)\boldsymbol{\Phi}^{\mathtt{T}}(\mathbf{T}_r)\boldsymbol{\Phi}(\mathbf{X})\boldsymbol{\Phi}^{\mathtt{T}}(\mathbf{X})\right\} \\ &+ (1-\alpha)\operatorname{Tr}\left\{\boldsymbol{\Phi}(\mathbf{T}_r)\boldsymbol{\Phi}^{\mathtt{T}}(\mathbf{T}_r)\hat{\boldsymbol{\Phi}}(\mathbf{Y})\hat{\boldsymbol{\Phi}}^{\mathtt{T}}(\mathbf{Y})\right\} \\ =& \operatorname{Tr}\left\{\boldsymbol{\Phi}(\mathbf{T}_r)\boldsymbol{\Phi}^{\mathtt{T}}(\mathbf{T}_r)\boldsymbol{\Phi}(\hat{\mathbf{G}})\right\} = \operatorname{Tr}\left\{\boldsymbol{\Phi}^{\mathtt{T}}(\mathbf{T}_r)\boldsymbol{\Phi}(\hat{\mathbf{G}})\boldsymbol{\Phi}(\mathbf{T}_r)\right\}\end{aligned}$$

with $\boldsymbol{\Phi}(\hat{\mathbf{G}}) = \alpha\boldsymbol{\Phi}(\mathbf{X})\boldsymbol{\Phi}^{\mathtt{T}}(\mathbf{X}) + (1-\alpha)\hat{\boldsymbol{\Phi}}(\mathbf{Y})\hat{\boldsymbol{\Phi}}^{\mathtt{T}}(\mathbf{Y})$. As a consequence, $\boldsymbol{\Upsilon}(\mathbf{W}_r)$ is maximized when $\mathbf{T}_r$ is selected such that $\boldsymbol{\Phi}(\mathbf{T}_r)$ contains the $4r$ first right eigenvectors of $\boldsymbol{\Phi}(\hat{\mathbf{G}})$. Notice that these structured eigenvectors exist since, from Property 1-8, it follows that

$$\boldsymbol{\Phi}^{\mathtt{T}}(\mathbf{T}_r)\boldsymbol{\Phi}(\hat{\mathbf{G}})\boldsymbol{\Phi}(\mathbf{T}_r) = \boldsymbol{\Phi}(\mathbf{D}) \quad \text{if and only if} \quad \mathbf{T}_r^{\mathtt{H}}\hat{\mathbf{G}}\mathbf{T}_r = \mathbf{D} \tag{A.14}$$

with $\mathbf{D}$ a diagonal matrix and thus, $\mathbf{T}_r$ contains the first $r$ right eigenvectors of $\hat{\mathbf{G}}$. Moreover, since $\hat{\boldsymbol{\Phi}}(\mathbf{Y}) = \boldsymbol{\Phi}(\hat{\mathbf{Y}})$ with $\hat{\mathbf{Y}} = \mathbf{X}\mathbf{X}^{+}\mathbf{Y}$, then $\hat{\mathbf{G}} = \alpha\mathbf{X}\mathbf{X}^{\mathtt{H}} + (1-\alpha)\hat{\mathbf{Y}}\hat{\mathbf{Y}}^{\mathtt{H}}$.

Finally, by minimizing the loss function: $\|\mathbf{T}_r - \mathbf{X}\mathbf{W}_r\|$, we get $\mathbf{W}_r = \mathbf{X}^{+}\mathbf{T}_r$.

2. The matrix function that we have to optimize can be rewritten as

$$\begin{aligned}\boldsymbol{\Upsilon}(\mathbf{W}_r) &= \operatorname{Tr}\left\{\boldsymbol{\Phi}(\mathbf{T}_r)\boldsymbol{\Phi}^{\mathtt{T}}(\mathbf{T}_r)\boldsymbol{\Phi}(\mathbf{G})\right\} \\ &= \operatorname{Tr}\left\{\boldsymbol{\Phi}^{\mathtt{T}}(\mathbf{W}_r)\boldsymbol{\Phi}^{\mathtt{T}}(\mathbf{X})\boldsymbol{\Phi}(\mathbf{G})\boldsymbol{\Phi}(\mathbf{X})\boldsymbol{\Phi}(\mathbf{W}_r)\right\}\end{aligned}$$

with $\boldsymbol{\Phi}(\mathbf{G}) = \alpha\boldsymbol{\Phi}(\mathbf{X})\boldsymbol{\Phi}^{\mathtt{T}}(\mathbf{X}) + (1-\alpha)\boldsymbol{\Phi}(\mathbf{Y})\boldsymbol{\Phi}^{\mathtt{T}}(\mathbf{Y})$ and $\mathbf{G} = \alpha\mathbf{X}\mathbf{X}^{\mathtt{H}} + (1-\alpha)\mathbf{Y}\mathbf{Y}^{\mathtt{H}}$.

From Property 1-8, it follows that

$$\boldsymbol{\Upsilon}(\mathbf{W}_r) = \operatorname{Tr}\left\{\boldsymbol{\Phi}^{\mathtt{T}}(\mathbf{W}_r)\big(\boldsymbol{\Phi}^{\mathtt{T}}(\mathbf{X})\boldsymbol{\Phi}(\mathbf{X})\big)^{1/2}\boldsymbol{\Phi}(\mathbf{H})\big(\boldsymbol{\Phi}^{\mathtt{T}}(\mathbf{X})\boldsymbol{\Phi}(\mathbf{X})\big)^{1/2}\boldsymbol{\Phi}(\mathbf{W}_r)\right\} \tag{A.15}$$

where $\mathbf{H} = (\mathbf{X}^{\mathtt{H}}\mathbf{X})^{-1/2}\mathbf{X}^{\mathtt{H}}\mathbf{G}\mathbf{X}(\mathbf{X}^{\mathtt{H}}\mathbf{X})^{-1/2}$. Note that $\mathbf{H}$ can be rewritten as (25). Thus, the result holds.

3. By minimizing the loss functions: $\|\mathbf{X} - \mathbf{T}_r\mathbf{P}_r^{\mathrm{H}}\|$ and $\|\mathbf{Y} - \mathbf{T}_r\mathbf{Q}_r^{\mathrm{H}}\|$.

4. $\hat{\mathbf{Y}} = \mathbf{\Pi}_{\mathbf{T}_r}\mathbf{Y} = \mathbf{T}_r\mathbf{T}_r^{+}\mathbf{Y} = \mathbf{X}\mathbf{W}_r\mathbf{T}_r^{+}\mathbf{Y}$.

## References


[1] S. Miron, J. Flamant, N.L. Bihan, P. Chainais, and D. Brie, Quaternions in signal and image processing: A comprehensive and objective overview, IEEE Signal Process. Magazine 40 (6) 2023 26-40.

[2] C. Zou, K.I. Kou, L. Dong, X. Zheng, and Y.Y. Tang, From grayscale to color: Quaternion linear regression for color face recognition, IEEE Access 7 (2019) 154131-154140.

[3] X. Xiao and Y. Zhou, Two-dimensional quaternion PCA and sparse PCA, IEEE Trans. Neural Netw. Learn. Syst. 30 (7) (2019) 2028-2042.

[4] J. Miao and K.I. Kou, Quaternion matrix regression for color face recognition, ArXiv (2020).

[5] M.T. El-Melegy, A.T. Kamal, K.F. Hussain, and H.M. El-Hawary, Face recognition by principal component regression using hypercomplex numbers, Assiut Univ. J. Multidiscip. Sci. Res. 51 (3) (2022) 268–278.

[6] M.T. El-Melegy and A.T. Kamal, Linear regression classification in the quaternion and reduced biquaternion domains, IEEE Signal Process. Letters 29 (2022) 469–473.

[7] M.T. El-Melegy, A.T. Kamal, K.F. Hussain, and H.M. El-Hawary, Classification by principal component regression in the real and hypercomplex domains, Arab. J. Sci. Eng. 48 (2023) 10099–10108.

[8] A.E. Stott, B.S. Dees, I. Kisil, and D.P. Mandic, A class of multidimensional NIPALS algorithms for quaternion and tensor partial least squares regression, Signal Process. 160 (2019) 316-327.

[9] M.M. Almeida, D. Mortari, R. Zanetti, and M. Akella, QuateRA: The quaternion regression algorithm, J. Guidance, Control, and Dynamics, 43 (9) 2020 1600-1616.

[10] J. Vía, D. Ramírez and I. Santamaría, Properness and widely linear processing of quaternion random vectors, IEEE Trans. Inform. Theory 56 (7) (2010) 3502-3515.

[11] T. Nitta, M. Kobayashi and D.P. Mandic, Hypercomplex widely linear estimation through the lens of underpinning geometry, IEEE Trans. Signal Process. 67 (15) (2019) 3985–3994.

[12] F. Zhang, Quaternions and matrices of quaternions, Linear Algebra Appl. 251 (1997) 21–57.

[13] W. Xiantao, Quaternion matrix and the re-nonnegative definite solutions to the quaternion matrix inverse problem $AX = B$, Math. J. Okayama Univ. 39 (1997) 61-69.

[14] Y. Tian and G.P.H. Styan, Some inequalities for sums of nonnegative definite matrices in quaternions, J. Inequalities App. 5 (2005) 449–458.

[15] N. Le Bihan and J. Mars, Singular value decomposition of quaternion matrices: a new tool for vector-sensor signal processing, Signal Process. 84 (2004) 1177–1199.

[16] Y. Tian, Universal factorization equalities for quaternion matrices and their applications, Math. J. Okayama Univ. 41 (1999) 45-62.

[17] M. Wei, Y. Li, F. Zhang, and J. Zhao, Quaternion Matrix Computations, Nova Science Publishers, New York, 2018.

[18] D. Zhang, T. Jiang, C. Jiang, and G. Wang, A complex structure-preserving algorithm for computing the singular value decomposition of a quaternion-matrix and its applications, Numer. Algorithms 95 (2024) 267–283.

[19] C. Cheong-Took and D.P. Mandic, Augmented second-order statistics of quaternion random signals, Signal Process. 91 (2011) 214–224.

[20] H. Wold, Estimation of principal components and related models by iterative least squares, in: P.R. Krishnajah (Ed.), Multivariate Analysis, Academic Press, New York, 1966, 391-420.

[21] R. Rosipal and N. Kramer, Overview and recent advances in partial least squares, in: C. Saunders, M. Grobelnik, S. Gunn, J. Shawe-Taylor (Eds), Subspace, Latent Structure and Feature Selection, Lecture Notes Computer Sci., Springer, Berlin, 2006, 34–51.

[22] S. de Jong, SIMPLS: an alternative approach to partial least squares regression, Chemome. Intell. Lab. Syst. 18 (1993) 251-263.

[23] S. de Jong and H.A.L. Kiers, Principal covariates regression. Part I. Theory, Chemome. Intell. Lab. Syst. 14 (1992) 155-164.

[24] J. Vía, D.P. Palomar and L.Vielva, Generalized likelihood ratios for testing the properness of quaternion Gaussian vectors, IEEE Trans. Signal Process. 59 (4) (2011) 1356-1370.

[25] P. Ginzberg and A.T. Walden, Testing for quaternion propriety, IEEE Trans. Signal Process. 59 (7) (2011) 3025-3034.

[26] S.C. Olhede, D. Ramírez, and P.J. Schreier, Detecting directionality in random fields using the monogenic signal, IEEE Trans. Inform. Theory 60 (10) (2014) 6491-6510.

[27] A. Sloin and A. Wiesel, Proper quaternion Gaussian graphical models, IEEE Trans. Signal Process. 62 (20) (2014) 5487-5496.

[28] E. Grassucci, D. Comminiello, and A. Uncini, An information-theoretic perspective on proper quaternion variational autoencoders, Entropy 23 (7) (2021) 856.

[29] A.C. Rencher, Multivariate Statistical Inference and Applications, Wiley, New York, 1998.

[30] C.E. Thomaz, FEI face database, 2012. http://fei.edu.br/∼cet/facedatabase.html.