Matrix psuedo-inverse and rank computation using Gram-Schmidt orthogonalization

Saurabh V. Pendse*[1]

[1] Department of Computer Science and Engineering
The Maharaja Sayajirao University of Baroda, Gujarat, India

June 15, 2010

Abstract

Matrix pseudo-inverse and rank computation are standard techniques in linear algebra and find application in many theoretical analyses as well as real world algorithms. The purpose of this technical note is to describe a general technique for computing matrix pseudo-inverse and rank of arbitrary non-square matrices using another technique from linear algebra - the Gram-Schmidt orthogonalization or QR factorization.

1 Notation

Vectors and matrices will be denoted in bold face, for example $\boldsymbol{\mu}$, $\boldsymbol{\beta}$, $\boldsymbol{X}$ etc. Given a matrix $\boldsymbol{X}$, its transpose will be denoted by $\boldsymbol{X}^T$, its pseudo-inverse will be denoted by $\boldsymbol{X}^-$ and its rank will be denoted by rank($\boldsymbol{X}$). The 2-norm of a vector $\boldsymbol{x}$ will be denoted by $||\boldsymbol{x}||_2 = +\sqrt{\boldsymbol{x}^T\boldsymbol{x}}$. The ijth element

*To whom correspo dence should be addressed. e-mail: sau2pen@gmail.com

of a matrix $\boldsymbol{X}$ will be denoted by X_{ij}. A vector or matrix of all zeros will be indicated by $\mathbf{0}$. Given a matrix $\boldsymbol{A}$, the Frobenius norm of $\boldsymbol{A}$ is given by $||\boldsymbol{A}||_{F_2} = \sqrt{\sum_{i,j} A_{ij}^2}$.

2 Introduction

We begin this section by formal definitions for matrix rank and pseudo-inverse. These computations find applications in many areas of statistics including linear least squares problems. We refer the reader to standard texts in linear algebra such as [1] and [2] for more details.

Definition 2.1. Matrix rank:
An $n \times p$ matrix $\boldsymbol{X}$ has rank q if there are at most q linearly independent columns in $\boldsymbol{X}$. Suppose the first q columns of $\boldsymbol{X}$ are linearly independent and let $\boldsymbol{X_q}$ be a $n \times q$ matrix containing the q linearly independent columns of $\boldsymbol{X}$. Then $\boldsymbol{X_q}\,\alpha = 0 \Rightarrow \alpha = 0$.

Definition 2.2. Matrix psuedoinverse:
We give the definition of psuedoinverse based on the solution of a linear least squares problem. Given an $n \times p$ matrix $\boldsymbol{X}$ and an arbitrary $n \times 1$ vector $\boldsymbol{y}$, suppose a $p \times 1$ vector $\boldsymbol{\beta}$ solves the linear least squares problem:

$$\min_{\boldsymbol{\beta}} \; (\boldsymbol{y} - \boldsymbol{X\beta})^T(\boldsymbol{y} - \boldsymbol{X\beta}) \tag{2.1}$$

Any solution $\hat{\boldsymbol{\beta}}$ for problem 2.1 will satisfy, the so called normal equation:

$$(\boldsymbol{X}^T\boldsymbol{X})\hat{\boldsymbol{\beta}} = \boldsymbol{X^T y} \tag{2.2}$$

If $n \geq p$ and $\text{rank}(\boldsymbol{X}) = p$, then 2.2 can be solved explicitly to yield the **unique** full rank least squares solution $\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X^T y}$. However, if $p > n$ or $\text{rank}(\boldsymbol{X}) = q < p$, then 2.2 has infinitely many solutions. A unique solution for this case can be obtained by choosing the one solution from among all solutions satisfying 2.2 that minimizes the squared norm $\boldsymbol{\beta}^T\boldsymbol{\beta}$ of $\boldsymbol{\beta}$. This solution can be written as:

$$\boldsymbol{\beta}^* = \boldsymbol{X}^- \boldsymbol{y} \tag{2.3}$$

The $p \times n$ matrix $\boldsymbol{X}^-$ is called the pseudo-inverse of $\boldsymbol{X}$.

The goal of this technical note is to derive explicit expressions for rank($\boldsymbol{X}$) and $\boldsymbol{X}^{-}$ using Gram-Schmidt orthogonalization or QR factorization.

3 Materials and Methods

We start with a detailed derivation of QR factorization followed by its application to rank and psuedo-inverse computation.

3.1 The QR factorization

In this section, we will describe the well known Gram-Schmidt orthogonalization process also known as QR factorization in linear algebra. Essentially the process involves re-expressing a set of vectors using an orthogonal set of basis vectors. Suppose we are given p vectors each of size $n \times 1$ arranged as a matrix $\boldsymbol{X} = [\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_p}]$. We also assume that each vector satisfies $\boldsymbol{x_i} \neq \boldsymbol{0}$.

Definition 3.1. Span of $\boldsymbol{X}$
The span of $\boldsymbol{X}$, denoted by Span($\boldsymbol{X}$) or Span($\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_p}$) is the set of all vectors that can be expressed as a linear combination of the columns of $\boldsymbol{X}$. Thus if $\boldsymbol{u} \in \text{Span}(\boldsymbol{X})$ then $\boldsymbol{u} = \sum_{i=1}^{p} \theta_i \boldsymbol{x_i}$ for some constants $\theta_1, \theta_2, \ldots, \theta_p$.

In Gram-Schmidt orthogonalization, we would like to derive a set of p vectors $\boldsymbol{z_1}, \boldsymbol{z_2}, \ldots, \boldsymbol{z_p}$ of size $n \times 1$ such that:

1.
$$\boldsymbol{x_i} \in \text{Span}(\boldsymbol{z_1}, \boldsymbol{z_2}, \ldots, \boldsymbol{z_i}) \quad \forall i \tag{3.1}$$

2. The vectors $\boldsymbol{z_1}, \boldsymbol{z_2}, \ldots, \boldsymbol{z_p}$ are orthogonal to each other:
$$\boldsymbol{z_i}^T \boldsymbol{z_j} = 0, \text{ if } i \neq j \tag{3.2}$$

Let us re-express all $\boldsymbol{x_i}$ in terms of the new orthogonal basis starting from $\boldsymbol{x_1}$. For the first vector $\boldsymbol{z_1}$ we can satisfy 3.1 by choosing:

$$\boldsymbol{x_1} = \boldsymbol{z_1} \tag{3.3}$$

Next, we express $\boldsymbol{x_2}$ in terms of $\boldsymbol{z_1}$ and $\boldsymbol{z_2}$ as follows:

$$\boldsymbol{x_2} = \alpha_{12}\, \boldsymbol{z_1} + \boldsymbol{z_2} \tag{3.4}$$

Premultiplying both sides of 3.4 by $\boldsymbol{z_1}^T$ and noting the orthogonal property 3.2 we see that:

$$\boldsymbol{z_1}^T \boldsymbol{x_2} = \alpha_{12}\, \boldsymbol{z_1}^T \boldsymbol{z_1} \tag{3.5}$$

A value of α_{12} that satisfies 3.5 can be defined as follows:

$$\alpha_{12} = \begin{cases} 0 & \text{if} \quad \boldsymbol{z_1} = \boldsymbol{0} \\ \frac{\boldsymbol{z_1}^T \boldsymbol{x_2}}{\boldsymbol{z_1}^T \boldsymbol{z_1}} & \text{if} \quad \boldsymbol{z_1} \neq \boldsymbol{0} \end{cases} \tag{3.6}$$

If $\boldsymbol{z_1} = \boldsymbol{0}$ then we choose $\alpha_{12} = 0$. Substituting α_{12} from 3.6 in 3.4 we get:

$$\boldsymbol{z_2} = \boldsymbol{x_2} - \{\alpha_{12}\}\, \boldsymbol{z_1} \tag{3.7}$$

Now suppose we proceed in a similar fashion deriving $\boldsymbol{z_1}, \boldsymbol{z_2}, \ldots, \boldsymbol{z_{i-1}}$. Let us express $\boldsymbol{x_i}$ in terms of $\boldsymbol{z_1}, \ldots, \boldsymbol{z_i}$ as follows:

$$\boldsymbol{x_i} = \sum_{k=1}^{i-1} \alpha_{ki}\, \boldsymbol{z_k} + \boldsymbol{z_i} \tag{3.8}$$

Premultiplying both sides by $\boldsymbol{z_k}^T$ for $k < i$ and noting the orthogonal property 3.2 we get:

$$\boldsymbol{z_k}^T \boldsymbol{x_i} = \alpha_{ki}\, \boldsymbol{z_k}^T \boldsymbol{z_k} \text{ where } k < i \tag{3.9}$$

A value of α_{ki} that satisfies 3.9 can be defined as follows:

$$\alpha_{ki} = \begin{cases} 0 & \text{if} \quad \boldsymbol{z_k} = \boldsymbol{0} \\ \frac{\boldsymbol{z_k}^T \boldsymbol{x_i}}{\boldsymbol{z_k}^T \boldsymbol{z_k}} & \text{if} \quad \boldsymbol{z_k} \neq \boldsymbol{0}, k < i \end{cases} \tag{3.10}$$

Again if $\boldsymbol{z_k} = \boldsymbol{0}$ then we choose $\alpha_{ki} = 0$. From 3.8 and 3.9 we can write:

$$\boldsymbol{z_i} = \boldsymbol{x_i} - \sum_{k=1}^{i-1} \{\alpha_{ki}\}\, \boldsymbol{z_k} \tag{3.11}$$

Equations 3.8 - 3.11 give the general recipe for computing $\boldsymbol{z_i}$ sequentially starting from $i = 1, \ldots, p$. If we define a $p \times p$ matrix $\boldsymbol{\Theta}$ with elements:

$$\Theta_{ki} = \begin{cases} \alpha_{ki} & \text{if } k < i \\ 1 & \text{if } k = i \\ 0 & \text{if } k > i \end{cases} \tag{3.12}$$

then we can write equation 3.8 in matrix form as follows:

$$\boldsymbol{X} = \boldsymbol{Z}\, \boldsymbol{\Theta} \tag{3.13}$$

where $\boldsymbol{Z} = [\boldsymbol{z_1}, \boldsymbol{z_2}, \ldots, \boldsymbol{z_p}]$ is a $n \times p$ matrix of orthogonal basis vectors.

Definition 3.2. Permutation Matrix
Let $\boldsymbol{e_i}$ be a $p \times 1$ vector of all 0's except in position i where it is 1. In other words, $\boldsymbol{e_i}$ is a unit vector. For example, if $p = 3$ then $\boldsymbol{e_2} = [0, 1, 0]^T$. Consider a matrix $\boldsymbol{P} = [\boldsymbol{e_{i_1}}, \boldsymbol{e_{i_2}}, \ldots, \boldsymbol{e_{i_p}}]$ where $i_1, i_2, \ldots, i_p$ is a permutation of $1, 2, \ldots, p$. Permutation matrices can be used to re-arrange columns of a matrix. For instance if $\boldsymbol{X}$ is a $n \times p$ matrix, then $\boldsymbol{X}\,\boldsymbol{P}$ is a permuted matrix with rearranged columns. The first column in $\boldsymbol{X}\,\boldsymbol{P}$ is the i_1th column in $\boldsymbol{X}$. The second column in $\boldsymbol{X}\,\boldsymbol{P}$ is the i_2th column in $\boldsymbol{X}$ and so on.

During the computation of $\boldsymbol{Z}$ it is possible that some $\boldsymbol{z_i}$ become $\boldsymbol{0}$. Suppose we keep a track of which $\boldsymbol{z_i}$'s become $\boldsymbol{0}$ and which do not. Suppose the non-zero $\boldsymbol{z_i}$ have indices $i_1, i_2, \ldots, i_q$ and the zero $\boldsymbol{z_i}$ have indices $i_{q+1}, \ldots, i_p$ where $i_1 < i_2 \ldots, i_q$. In the matrix product 3.13 the rows in $\boldsymbol{\Theta}$ corresponding to indices $i_{q+1}, \ldots, i_p$ drop out since the corresponding columns in $\boldsymbol{Z}$ are $\boldsymbol{0}$. Suppose $\boldsymbol{Z}_q$ is the $n \times q$ matrix with columns $\boldsymbol{z_{i_1}}, \boldsymbol{z_{i_2}}, \ldots, \boldsymbol{z_{i_q}}$ and $\boldsymbol{\Theta}_q$ is the $q \times p$ matrix made up of the rows $i_1, i_2, \ldots, i_q$ from $\boldsymbol{\Theta}$ then we can write 3.13 as follows:

$$\boldsymbol{X} = \boldsymbol{Z}_q\,\boldsymbol{\Theta}_q \tag{3.14}$$

In matrix $\boldsymbol{\Theta}_q$, the ljth entry is the contribution of vector $\boldsymbol{z_{i_l}}$ to a linear expression of vector $\boldsymbol{x_j}$ in terms of the columns of $\boldsymbol{Z}_q$. It is clear from 3.8 that the the l-i_lth term of $\boldsymbol{\Theta}_q$, i.e., the contribution of $\boldsymbol{z_{i_l}}$ to x_{i_l} will be 1 i.e, $\Theta_q(l, i_l) = 1$. Also, $\boldsymbol{z_{i_j}}$ does not make a contribution to x_{i_l} if $i_j > i_l$. Therefore $\Theta_q(j, i_l) = 0$ for $j > l$. Let us define a $p \times p$ permutation matrix $\boldsymbol{P} = [\boldsymbol{e_{i_1}}, \ldots, \boldsymbol{e_{i_p}}]$ and post multiply both sides of equation 3.14 by $\boldsymbol{P}$. This gives

$$\boldsymbol{X}\,\boldsymbol{P} = \boldsymbol{Z}_q\,[\boldsymbol{\Theta}_q\,\boldsymbol{P}] \tag{3.15}$$

By the property of permutation matrices, the lth column in $[\boldsymbol{\Theta}_q\,\boldsymbol{P}]$ is the i_lth column in $\boldsymbol{\Theta}_q$. Thus $\Theta_q(l, i_l) = [\boldsymbol{\Theta}_q\,\boldsymbol{P}](l, l) = 1$ and $\Theta_q(j, i_l) = [\boldsymbol{\Theta}_q\,\boldsymbol{P}](j, l) = 0$ if $j > l$. Therefore the left $q \times q$ submatrix of $[\boldsymbol{\Theta}_q\boldsymbol{P}]$ is upper triangular with 1's on the diagonal and hence is non-singular. Let us partition

$$[\boldsymbol{\Theta}_q\boldsymbol{P}] = [\boldsymbol{M_1}, \boldsymbol{M_2}] \tag{3.16}$$

where $\boldsymbol{M_1}$ is $q \times q$ and non-singular. $\boldsymbol{M_2}$ is a $q \times (p - q)$ matrix. Thus we can write

$$\boldsymbol{X}\,\boldsymbol{P} = \boldsymbol{Z}_q[\boldsymbol{M_1}, \boldsymbol{M_2}] \tag{3.17}$$

If $\boldsymbol{\Lambda}$ is a diagonal matrix with elements $\Lambda(j,j) = ||\boldsymbol{z_{i_j}}||_2$ then we can re-write 3.18 as:

$$\boldsymbol{XP} = \boldsymbol{Z}_q \boldsymbol{\Lambda}^{-1} \boldsymbol{\Lambda}[\boldsymbol{M_1}, \boldsymbol{M_2}] \tag{3.18}$$

Now define the $n \times q$ matrix $\boldsymbol{Q}$ and $q \times p$ matrix $\boldsymbol{R}$ as follows:

$$\boldsymbol{Q} = \boldsymbol{Z}_q \boldsymbol{\Lambda}^{-1} \tag{3.19}$$
$$\boldsymbol{R} = [\boldsymbol{R_1}, \boldsymbol{R_2}] = \boldsymbol{\Lambda}[\boldsymbol{M_1}, \boldsymbol{M_2}] \tag{3.20}$$
$$\boldsymbol{R_1} = \boldsymbol{\Lambda M_1} \tag{3.21}$$
$$\boldsymbol{R_2} = \boldsymbol{\Lambda M_2} \tag{3.22}$$

By construction $\boldsymbol{Q}$ is an orthonormal matrix satisfying $\boldsymbol{Q}^T\boldsymbol{Q} = \boldsymbol{I}_q$. Thus we finally get the Q-R factorization of $\boldsymbol{X}$ as follows:

$$\boxed{\boldsymbol{XP} = \boldsymbol{Q}\,[\boldsymbol{R_1}, \boldsymbol{R_2}] = \boldsymbol{Q}\,\boldsymbol{R}} \tag{3.23}$$

where $\boldsymbol{R_1}$ is a square and non-singular matrix of size $q \times q$ and $\boldsymbol{R_2}$ is a $q \times (p-q)$ matrix. $\boldsymbol{P}$ is a $p \times p$ permutation matrix which permutes the p columns of $\boldsymbol{X}$ such that the first q columns are linearly independent.

3.2 Application to rank computation

Since $\boldsymbol{R_1}$ is invertible we can write 3.23 as:

$$\boldsymbol{XP} = \boldsymbol{Q}\boldsymbol{R_1}[\boldsymbol{I}_q, \boldsymbol{R_1}^{-1}\boldsymbol{R_2}] \tag{3.24}$$

In other words, the last $(p-q)$ columns of $\boldsymbol{XP}$ can be expressed as a linear combination of the first q columns. Morever, the first q columns are linearly independent since:

$$\boldsymbol{Q}\boldsymbol{R_1}\boldsymbol{\alpha} = \boldsymbol{0} \tag{3.25}$$
$$\Rightarrow \boldsymbol{R_1}\boldsymbol{\alpha} = \boldsymbol{0} \qquad \text{pre multiplying both sides by } \boldsymbol{Q}^T \tag{3.26}$$
$$\Rightarrow \boldsymbol{\alpha} = \boldsymbol{0} \qquad \text{since } \boldsymbol{R_1} \text{ is non-singular} \tag{3.27}$$

Therefore, $\boldsymbol{XP}$ has rank q. Since $\boldsymbol{XP}$ has the permuted columns of $\boldsymbol{X}$, the rank of $\boldsymbol{X}$ and $\boldsymbol{XP}$ is the same. In other words, $\text{rank}(\boldsymbol{X}) = q$, the size of square matrix $\boldsymbol{R_1}$.

3.3 Application to psuedo-inverse computation

Suppose $\boldsymbol{\beta^*}$ is the minimum-norm solution to the least squares problem 2.1. Since $\boldsymbol{P}$ is a permutation matrix, it satisfies $\boldsymbol{P}\boldsymbol{P}^T = \boldsymbol{P}^T\boldsymbol{P} = \boldsymbol{I}_p$. Thus we can write 3.23 as:

$$\boldsymbol{X} = \boldsymbol{Q}\,[\boldsymbol{R_1}, \boldsymbol{R_2}]\boldsymbol{P}^T \tag{3.28}$$

Substituting 3.28 into the normal equations 2.2 we get:

$$\boldsymbol{P}\begin{pmatrix} \boldsymbol{R_1}^T\boldsymbol{R_1} & \boldsymbol{R_1}^T\boldsymbol{R_2} \\ \boldsymbol{R_2}^T\boldsymbol{R_1} & \boldsymbol{R_2}^T\boldsymbol{R_2} \end{pmatrix}\boldsymbol{P}^T\boldsymbol{\beta^*} = \boldsymbol{P}\begin{pmatrix} \boldsymbol{R_1}^T \\ \boldsymbol{R_2}^T \end{pmatrix}\boldsymbol{Q}^T\boldsymbol{y} \tag{3.29}$$

Let us parameterize $\boldsymbol{P}^T\boldsymbol{\beta^*}$ as follows:

$$\boldsymbol{P}^T\boldsymbol{\beta^*} = \begin{pmatrix} \boldsymbol{h_1} \\ \boldsymbol{h_2} \end{pmatrix} = \boldsymbol{h} \tag{3.30}$$

where $\boldsymbol{h_1}$ is a $q \times 1$ vector and $\boldsymbol{h_2}$ is $(p-q) \times 1$ vector. Substituting 3.30 in 3.29 and simplifying we get:

$$(\boldsymbol{R_1}^T\boldsymbol{R_1})\boldsymbol{h_1} + (\boldsymbol{R_1}^T\boldsymbol{R_2})\boldsymbol{h_2} = \boldsymbol{R_1}^T\boldsymbol{Q}^T\boldsymbol{y} \tag{3.31}$$

$$(\boldsymbol{R_2}^T\boldsymbol{R_1})\boldsymbol{h_1} + (\boldsymbol{R_2}^T\boldsymbol{R_2})\boldsymbol{h_2} = \boldsymbol{R_2}^T\boldsymbol{Q}^T\boldsymbol{y} \tag{3.32}$$

Only the first equation is independent since the second can be derived from the first by pre multiplying both sides by $\boldsymbol{R_2}^T\boldsymbol{R_1}^{-T}$. From 3.31 we can solve for $\boldsymbol{h_1}$ to get:

$$\boldsymbol{h_1} = (\boldsymbol{R_1}^T\boldsymbol{R_1})^{-1}\boldsymbol{R_1}^T\boldsymbol{Q}^T\boldsymbol{y} - (\boldsymbol{R_1}^T\boldsymbol{R_1})^{-1}(\boldsymbol{R_1}^T\boldsymbol{R_2})\boldsymbol{h_2} \tag{3.33}$$

$$\tag{3.34}$$

Suppose

$$\boldsymbol{c} = (\boldsymbol{R_1}^T\boldsymbol{R_1})^{-1}\boldsymbol{R_1}^T\boldsymbol{Q}^T\boldsymbol{y} \tag{3.35}$$

$$\boldsymbol{D} = (\boldsymbol{R_1}^T\boldsymbol{R_1})^{-1}(\boldsymbol{R_1}^T\boldsymbol{R_2}) \tag{3.36}$$

From 3.35 and 3.33 we can write:

$$\boldsymbol{h_1} = \boldsymbol{c} - \boldsymbol{D}\boldsymbol{h_2} \tag{3.37}$$

We would like to solve for $\boldsymbol{h_1}$ and $\boldsymbol{h_2}$ such that the norm of $\boldsymbol{\beta^*}$ is minimized. Now $\boldsymbol{\beta^*}^T\boldsymbol{\beta^*} = \boldsymbol{\beta^*}^T\boldsymbol{P}\boldsymbol{P}^T\boldsymbol{\beta^*} = \begin{pmatrix} \boldsymbol{h_1}^T & \boldsymbol{h_2}^T \end{pmatrix}\begin{pmatrix} \boldsymbol{h_1} \\ \boldsymbol{h_2} \end{pmatrix} = \boldsymbol{h_1}^T\boldsymbol{h_1} + \boldsymbol{h_2}^T\boldsymbol{h_2}$.

Therefore minimizing the norm of $\boldsymbol{\beta^*}$ is equivalent to solving the following optimization problem:

$$\begin{aligned} &\text{minimize} \quad \boldsymbol{h_1}^T\boldsymbol{h_1} + \boldsymbol{h_2}^T\boldsymbol{h_2} \\ &\text{suject to:} \quad \boldsymbol{h_1} = \boldsymbol{c} - \boldsymbol{D}\boldsymbol{h_2} \end{aligned} \tag{3.38}$$

Substituting $\boldsymbol{h_1}$ from 3.37 into 3.38 we can see that the objective function to be minimized as a function of $\boldsymbol{h_2}$ can be written as:

$$f(\boldsymbol{h_2}) = (\boldsymbol{c} - \boldsymbol{D}\boldsymbol{h_2})^T(\boldsymbol{c} - \boldsymbol{D}\boldsymbol{h_2}) + \boldsymbol{h_2}^T\boldsymbol{h_2} \tag{3.39}$$

Differentiating 3.39 w.r.t $\boldsymbol{h_2}$ and equating to $\mathbf{0}$ we get the necessary conditions for optimality:

$$\frac{\partial}{\partial \boldsymbol{h_2}} f(\boldsymbol{h_2}) = -2\boldsymbol{D}^T\boldsymbol{c} + 2\boldsymbol{D}^T\boldsymbol{D}\boldsymbol{h_2} + 2\boldsymbol{h_2} = \mathbf{0} \tag{3.40}$$

Note that the Hessian of 3.39 w.r.t $\boldsymbol{h_2}$ is $2(\boldsymbol{D}^T\boldsymbol{D} + \boldsymbol{I}_{(p-q)})$ which is positive definite and hence solution of 3.40 will actually minimize the objective 3.39 as desired. $\boldsymbol{h_1}$ and $\boldsymbol{h_2}$ can be found as the solution of the following system of linear equations:

$$\boldsymbol{D}^T\boldsymbol{D}\boldsymbol{h_2} + \boldsymbol{h_2} = \boldsymbol{D}^T\boldsymbol{c} \tag{3.41}$$

$$\boldsymbol{h_1} = \boldsymbol{c} - \boldsymbol{D}\boldsymbol{h_2} \tag{3.42}$$

Substituting $\boldsymbol{D}\boldsymbol{h_2}$ from 3.42 in 3.41 we get:

$$\boldsymbol{h_2} = \boldsymbol{D}^T\boldsymbol{h_1} \tag{3.43}$$

Substituting 3.43 into 3.42 and solving for $\boldsymbol{h_1}$ and $\boldsymbol{h_2}$ we get:

$$\boldsymbol{h_1} = \left(\boldsymbol{D}\boldsymbol{D}^T + \boldsymbol{I}_q\right)^{-1}\boldsymbol{c} \tag{3.44}$$

$$\boldsymbol{h_2} = \boldsymbol{D}^T\left(\boldsymbol{D}\boldsymbol{D}^T + \boldsymbol{I}_q\right)^{-1}\boldsymbol{c} \tag{3.45}$$

Substituting $\boldsymbol{D}$ from 3.36 we get:

$$\left(\boldsymbol{D}\boldsymbol{D}^T + \boldsymbol{I}_q\right) = \boldsymbol{R_1}^{-1}\left(\boldsymbol{R_1}\boldsymbol{R_1}^T + \boldsymbol{R_2}\boldsymbol{R_2}^T\right)\boldsymbol{R_1}^{-T} \tag{3.46}$$

From 3.46, 3.35, 3.44 and 3.45 we can solve for $\boldsymbol{h_1}$ and $\boldsymbol{h_2}$ to get:

$$\boldsymbol{h_1} = \boldsymbol{R_1}^T \left(\boldsymbol{R_1}\boldsymbol{R_1}^T + \boldsymbol{R_2}\boldsymbol{R_2}^T\right)^{-1} \boldsymbol{Q}^T\boldsymbol{y} \tag{3.47}$$

$$\boldsymbol{h_2} = \boldsymbol{R_2}^T \left(\boldsymbol{R_1}\boldsymbol{R_1}^T + \boldsymbol{R_2}\boldsymbol{R_2}^T\right)^{-1} \boldsymbol{Q}^T\boldsymbol{y} \tag{3.48}$$

First note that $\boldsymbol{R_1}$ is invertible and so $\boldsymbol{R_1}\boldsymbol{R_1}^T$ is a symmetric and positive definite matrix. Thus for any $\boldsymbol{x} \neq 0$, $\boldsymbol{x}^T\boldsymbol{R_1}\boldsymbol{R_1}^T\boldsymbol{x} = ||\boldsymbol{R_1}^T\boldsymbol{x}||_2^2 > 0$. Therefore $\left(\boldsymbol{R_1}\boldsymbol{R_1}^T + \boldsymbol{R_2}\boldsymbol{R_2}^T\right)$ is also a symmetric and positive definite matrix. This is because if $\boldsymbol{x} \neq 0$ then $\boldsymbol{x}^T\left(\boldsymbol{R_1}\boldsymbol{R_1}^T + \boldsymbol{R_2}\boldsymbol{R_2}^T\right)\boldsymbol{x} = ||(\boldsymbol{R_1}^T\boldsymbol{x})||_2^2 + ||(\boldsymbol{R_2}^T\boldsymbol{x})||_2^2 > 0$. Thus $\left(\boldsymbol{R_1}\boldsymbol{R_1}^T + \boldsymbol{R_2}\boldsymbol{R_2}^T\right)$ is also invertible. From 3.47, 3.23 and 3.30 we get:

$$\boldsymbol{P}^T\boldsymbol{\beta^*} = \begin{pmatrix}\boldsymbol{h_1}\\ \boldsymbol{h_2}\end{pmatrix} = \begin{pmatrix}\boldsymbol{R_1}^T(\boldsymbol{R}\boldsymbol{R}^T)^{-1}\boldsymbol{Q}^T\boldsymbol{y}\\ \boldsymbol{R_2}^T(\boldsymbol{R}\boldsymbol{R}^T)^{-1}\boldsymbol{Q}^T\boldsymbol{y}\end{pmatrix} = \boldsymbol{R}^T(\boldsymbol{R}\boldsymbol{R}^T)^{-1}\boldsymbol{Q}^T\boldsymbol{y} \tag{3.49}$$

We premultiply both sides of 3.49 by $\boldsymbol{P}$ and note that $\boldsymbol{P}\boldsymbol{P}^T = \boldsymbol{I}_p$ to get:

$$\boxed{\boldsymbol{\beta^*} = \boldsymbol{P}\boldsymbol{R}^T(\boldsymbol{R}\boldsymbol{R}^T)^{-1}\boldsymbol{Q}^T\boldsymbol{y}} \tag{3.50}$$

Comparing the minimum-norm solution 3.50 with 2.3 we see that the matrix pseudo inverse of $\boldsymbol{X}$ is given by:

$$\boldsymbol{\beta^*} = \boldsymbol{X}^-\boldsymbol{y} \tag{3.51}$$

$$\boxed{\boldsymbol{X}^- = \boldsymbol{P}\boldsymbol{R}^T(\boldsymbol{R}\boldsymbol{R}^T)^{-1}\boldsymbol{Q}^T} \tag{3.52}$$

3.4 Inverse of a square matrix

It is clear from 3.52 that pseudo-inverse computation involves computing the inverse of the $q \times q$ matrix $\boldsymbol{R}\boldsymbol{R}^T$. As discussed below 3.47, this matrix is square and invertible. In this section, we will see how we can use an additional QR factorization to compute this inverse. We repeat the Gram-Schmidt orthogonalization on matrix $\boldsymbol{R}\boldsymbol{R}^T$ to get:

$$\boldsymbol{R}\boldsymbol{R}^T = \boldsymbol{Q_1}\boldsymbol{R_1} \tag{3.53}$$

Note that the permutation matrix such as in 3.24 is an identity matrix since $\boldsymbol{R}\boldsymbol{R}^T$ is invertible and hence has full rank q. $\boldsymbol{Q_1}$ is a $q \times q$ orthonormal matrix satisfying $\boldsymbol{Q_1}^T\boldsymbol{Q_1} = \boldsymbol{I}_q$ and $\boldsymbol{R_1}$ is a $q \times q$ non-singular and upper triangular

matrix (i.e., entries below the diagonal are 0). Inverting both sides of 3.53 we get:

$$(\boldsymbol{R}\boldsymbol{R}^T)^{-1} = (\boldsymbol{Q_1}\boldsymbol{R_1})^{-1} = \boldsymbol{R_1}^{-1}\boldsymbol{Q_1}^{-1} = \boldsymbol{R_1}^{-1}\boldsymbol{Q_1}^T \tag{3.54}$$

The last equality in 3.54 follows from the fact that $\boldsymbol{Q_1}$ is orthonormal and hence the inverse of $\boldsymbol{Q_1}$ is just the transpose $\boldsymbol{Q_1}^T$. If we can compute the inverse of an upper triangular matrix $\boldsymbol{R_1}$ then we can compute the inverse of $\boldsymbol{R}\boldsymbol{R}^T$. The next section describes how to compute the inverse of an upper triangular matrix.

3.5 Inverse of an upper triangular matrix

Suppose $\boldsymbol{G}$ is an upper triangular matrix with inverse $\boldsymbol{H}$. Let G_{ij} denote the i-jth element of $\boldsymbol{G}$ and H_{ij} denote the i-jth element of $\boldsymbol{H}$ then we can write in matrix form:

$$\begin{pmatrix} G_{11} & G_{12} & \dots & G_{1q} \\ 0 & G_{22} & \dots & G_{2q} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & G_{qq} \end{pmatrix} \begin{pmatrix} H_{11} & H_{12} & \dots & H_{1q} \\ H_{21} & H_{22} & \dots & H_{2q} \\ \vdots & \vdots & \ddots & \vdots \\ H_{q1} & H_{q2} & \dots & H_{qq} \end{pmatrix} = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{pmatrix} \tag{3.55}$$

Let I_{ik} be the i-kth element of $\boldsymbol{I}_q$. Equating the i-kth element on both sides we can write:

$$\sum_{j=1}^{q} G_{ij}\, H_{jk} = I_{ik} \tag{3.56}$$

Since $\boldsymbol{G}$ is upper triangular, $G_{ij} = 0$ if $i > j$. Thus we can re-write 3.56 as:

$$G_{ii}H_{ik} + \sum_{j=i+1}^{q} G_{ij}H_{jk} = I_{ik} \tag{3.57}$$

The summation term in the above equation includes only the rows of $\boldsymbol{H}$ below row i. When $i = q$ then 3.57 can be used to solve for the last row of $\boldsymbol{H}$. When $i = (q-1)$, the summation term will involve only rows below $(q-1)$ i.e, row q. Thus when $i = (q-1)$ we can solve for the $(q-1)$th row of $\boldsymbol{H}$. When $i = (q-2)$ we can solve for the $(q-2)$th row of $\boldsymbol{H}$ since rows $(q-1)$ and q of $\boldsymbol{H}$ are already known at this point. Continuing in this fashion for $i = (q-3), \dots, 1$ we can compute the matrix $\boldsymbol{H}$.

3.6 Improving computational stability

We use a constant threshold to detect zero vectors $\boldsymbol{z_i}$ during the QR factorization process via a condition such as $||\boldsymbol{z_i}||_2 < \varepsilon$. If some vectors $\boldsymbol{x_i}$ are scaled differently than others then the use of this constant threshold ε is not appropriate. This problem can be easily addressed by re-scaling each column of $\boldsymbol{X}$ before starting calculations. Assuming $\boldsymbol{x_i} \neq 0$, suppose $\boldsymbol{D}$ is a diagonal matrix such that $D_{ii} = ||\boldsymbol{x_i}||_2$ then we define a rescaled input matrix as:

$$\boldsymbol{X_N} = \boldsymbol{X}\boldsymbol{D}^{-1} \tag{3.58}$$

QR factorization is now applied to $\boldsymbol{X_N}$ to get as per 3.23

$$\boldsymbol{X_N}\boldsymbol{P_N} = \boldsymbol{Q_N}\boldsymbol{R_N} \tag{3.59}$$

Substituting 3.58 into 3.59 and solving for $\boldsymbol{X}$ we get:

$$\boldsymbol{X} = \boldsymbol{Q_N}\boldsymbol{R_N}\boldsymbol{P_N}^T\boldsymbol{D} \tag{3.60}$$

Define a new matrix $\boldsymbol{R^*_N}$ as follows:

$$\boldsymbol{R^*_N} = \boldsymbol{R_N}\boldsymbol{P_N}^T\boldsymbol{D} \tag{3.61}$$

Thus we can write $\boldsymbol{X}$ as:

$$\boldsymbol{X} = \boldsymbol{Q_N}\boldsymbol{R^*_N} \tag{3.62}$$

Note that the permutation matrix is identity in 3.62. Invoking the definition of pseudo-inverse from 3.52 we get:

$$\boxed{\boldsymbol{X}^- = \boldsymbol{R^*_N}^T\left(\boldsymbol{R^*_N}\boldsymbol{R^*_N}^T\right)^{-1}\boldsymbol{Q_N}^T} \tag{3.63}$$

4 Experiments and Results

In this section, we verify the accuracy of our rank and pseudo-inverse computations using carefully simulated matrices $\boldsymbol{X}$ and compare our results to those from Matlab (`www.mathworks.com`).

1. We generate random matrices $\boldsymbol{X}$ as follows:

$$\boldsymbol{X} = [\boldsymbol{A_1}\boldsymbol{B_1}, \boldsymbol{A_1}\boldsymbol{B_2}, \boldsymbol{A_2}\boldsymbol{B_3}] \tag{4.1}$$

where $\boldsymbol{A_1}$ is a $n \times q_1$ matrix, $\boldsymbol{B_1}$ is a $q_1 \times p_1$ matrix, $\boldsymbol{B_2}$ is a $q_1 \times p_2$ matrix, $\boldsymbol{A_2}$ is a $n \times q_2$ matrix and $\boldsymbol{B_3}$ is a $q_2 \times p_3$ matrix. Matrix $\boldsymbol{B_2}$ is generated by selecting a set of columns from $\boldsymbol{B_1}$. If $\mathcal{I}$ is an index set containing the indices of $p_2 < p_1$ columns from $\boldsymbol{B_1}$ then $\boldsymbol{B_2} = \boldsymbol{B_1}(:, \mathcal{I})$. We also randomly choose $n, q_1, p_1, p_2, q_2, p_3$ from the set $\{2, 3, \ldots, 200\}$. This construction will ensure that the true rank of $\boldsymbol{X}$ will often be smaller than $\min(n, p)$ making the rank and pseudo-inverse computation interesting problems.

2. For each random $\boldsymbol{X}$, we calculate $\text{rank}(\boldsymbol{X})$ using our algorithm. Let $\text{rank}_{matlab}(\boldsymbol{X})$ be the value returned by the function `rank` in Matlab. We compute Δ_{rank}, the absolute difference in rank computed by our algorithm and by the Matlab function `rank` as a measure of accuracy:

$$\Delta_{rank} = |\text{rank}(\boldsymbol{X}) - \text{rank}_{matlab}(\boldsymbol{X})| \tag{4.2}$$

where $|.|$ denotes absolute value.

3. Similarly for each $\boldsymbol{X}$, we compute $\boldsymbol{X}^-$ using our algorithm. Let $\boldsymbol{X^-_{matlab}}$ be the pseudo-inverse computed by the function `pinv` in Matlab. We compute Δ_{pinv}, the Frobenius norm of the difference between the pseudo-inverse computed by our algorithm and by the Matlab function `pinv` as a measure of accuracy:

$$\Delta_{pinv} = ||\boldsymbol{X}^- - \boldsymbol{X^-_{matlab}}||_{F_2} \tag{4.3}$$

We generated 50 independent samples of matrices $\boldsymbol{X}$ of various random sizes and computed the rank and pseudo-inverse performance metrics as described above.

- Fig. 1 shows the number of rows and columns in each of the 50 random matrices $\boldsymbol{X}$ as well as $\text{rank}(\boldsymbol{X})$ computed using our algorithm as well as the function `rank` in Matlab. It can be seen that both our algorithm and the function `rank` produce identical results in all 50 samples of $\boldsymbol{X}$.

- Fig. 2 shows Δ_{rank} and Δ_{pinv} computations for the 50 random $\boldsymbol{X}$. We found that Δ_{rank} was 0 for all 50 samples while Δ_{pinv} was $< 4 \times 10^{-5}$ for all 50 samples. The small difference on the order of 10^{-5} in the Frobenius norm of the difference in pseudo-inverse between our algorithm and Matlab can be explained by round off errors during computations.

Both Fig. 1 and Fig. 2 suggest that our derivation and algorithm implementation for computing $\text{rank}(\boldsymbol{X})$ and $\boldsymbol{X}^-$ is accurate.

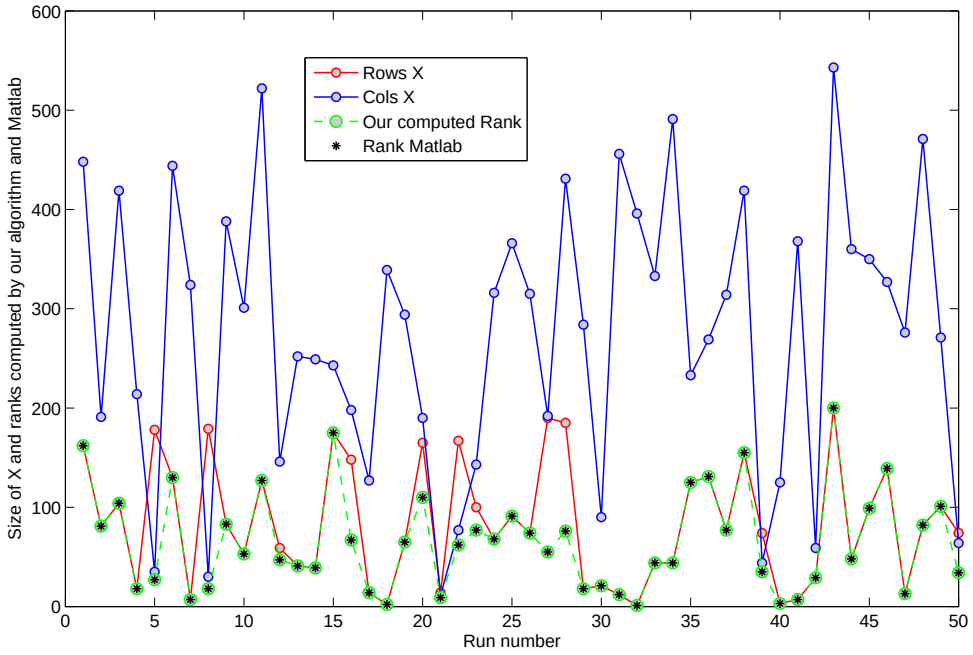

Figure 1: For each simulated $\boldsymbol{X}$, figure shows the number of rows and columns in $\boldsymbol{X}$ as well as the rank($\boldsymbol{X}$) computed using our algorithm and by Matlab's function 'rank'.

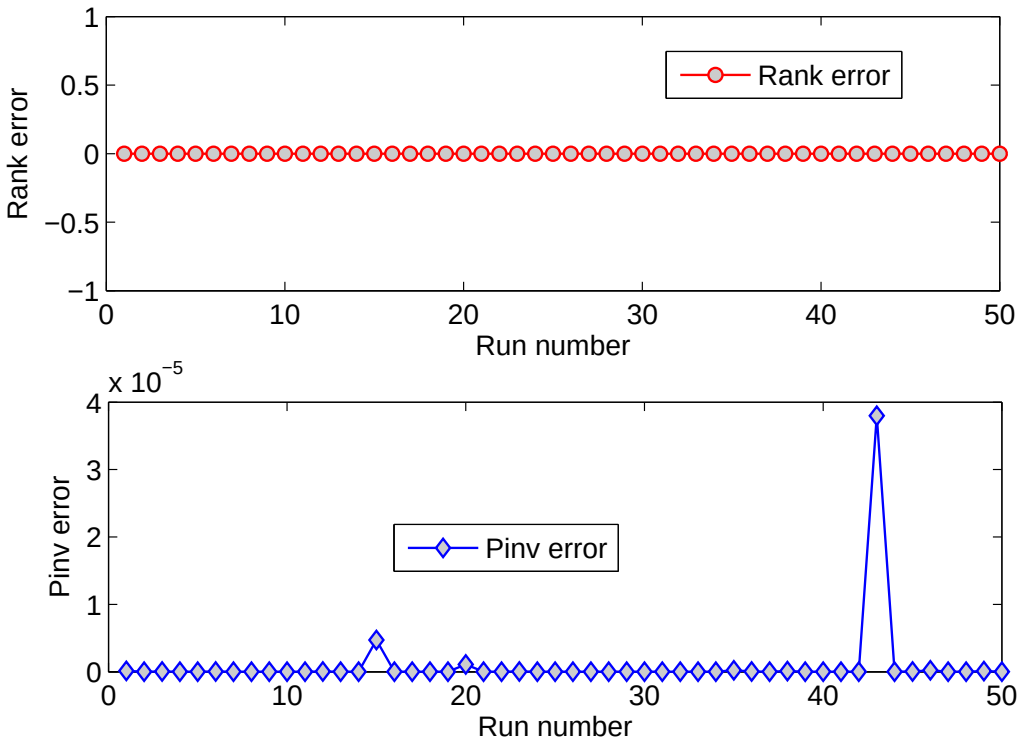

Figure 2: For each simualted $\boldsymbol{X}$, we computed the rank and pseudo inverse using our algorithm and the functions 'rank' and 'pinv' in Matlab. This figure shows the accuracy measures Δ_{rank} and Δ_{pinv} as defined in 4.2 and 4.3 for each simulation.

5 Discussion

We first derived in detail the Gram-Schmidt orthogonalization or the QR factorization equations. We then applied this factorization to the problem of computing the rank and pseudo-inverse of an arbitrary non-square matrix. In practice, the condition $\boldsymbol{z_k} \neq \boldsymbol{0}$ during QR factorization is checked using an inequality such as $||\boldsymbol{z_k}||_2 \leq \varepsilon$ where ε is a small quantity such as 10^{-6}. To enable the use of a constant threshold ε in this condition, we described a technique where we re-scale each column of $\boldsymbol{X}$ before applying QR factorization. The pseudo-inverse computation can be re-written in terms of this re-scaled matrix as described in 3.63. Also note that when the $n \times p$ matrix $\boldsymbol{X}$ has rank p i.e., $\boldsymbol{X}$ is full rank then $\text{rank}(\boldsymbol{X}) = p$ and the permutation matrix $\boldsymbol{P} = \boldsymbol{I}_p$. Since $\boldsymbol{R}$ is invertible in this case, equation 3.51 simplifies to the well known unique least-squares solution $\boldsymbol{\beta}^* = \boldsymbol{R}^{-1}\boldsymbol{Q}^T\boldsymbol{y}$.

The matrix pseudo-inverse computation enables us to compute the minimum norm solution to the least squares problem when $\boldsymbol{X}$ is not full rank. Experiments on randomly generated matrices $\boldsymbol{X}$ of various sizes and comparison of our results with the results of functions `rank` and `pinv` in Matlab verify the correctness in derivation and implementation of our algorithm.

6 Conclusion

We derived in detail the computation of rank and pseudo-inverse of a matrix using the QR factorization method. The resulting algorithm is fast and produces accurate results as evidenced by comparison with results from Matlab.

References

[1] Golub, G. H. and Van Loan, C. F. *Matrix Computations*, 3rd edition, The Johns Hopkins University Press (1996).

[2] Demmel, J. W. *Applied Numerical Linear Algebra*, Society for Industrial and Applied Mathematics (1997).